



Universidad
Carlos III de Madrid

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Ingeniería Sistemas y Automática

ESTUDIO DE LA HERRAMIENTA FRAMSTICKS 3D PARA LA SIMULACIÓN DE ROBOTS EVOLUTIVOS

TRABAJO FIN DE GRADO

Autora: Marina Cabañas Viso

Tutora: Concepción Alicia Monje Micharet

Directora: Tamara Ramos Cambero

Leganés, Marzo del 2015

ESTUDIO DE LA HERRAMIENTA FRAMSTICKS PARA LA SIMULACIÓN DE ROBOTS EVOLUTIVOS

Por

Marina Cabañas Viso

Presentado en la

ESCUELA POLITÉCNICA SUPERIOR

De la

UNIVERSIDAD CARLOS III DE MADRID

Para la obtención del

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Directora de Trabajo Fin de Grado

Dña. Tamara Ramos Cambero

Tutora de Trabajo Fin de Grado

Dña. Concepción Alicia Monje Micharet

Madrid, 5 de Marzo 2015

TRABAJO FIN DE GRADO

**Título: ESTUDIO DE LA HERRAMIENTA FRAMSTICKS 3D PARA LA
SIMULACIÓN DE ROBOTS EVOLUTIVOS**

Autora: Marina Cabañas Viso

Directora: Tamara Ramos Cambero

Tutora: Concepción Alicia Monje Micharet

EL TRIBUNAL

Presidente: Jose Carlos Castillo Montoya

Vocal: Verónica González Pérez

Secretario: Luis García Sánchez

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 5 de Marzo de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	IX
ÍNDICE DE ECUACIONES	X
AGRADECIMIENTOS	1
RESUMEN	2
ABSTRACT	3
CAPÍTULO 1.....	4
INTRODUCCIÓN Y OBJETIVOS	4
1.1 INTRODUCCIÓN	4
1.2 OBJETIVOS.....	5
1.3 ESTRUCTURA DE LA MEMORIA.....	6
CAPÍTULO 2	7
FRAMSTICKS	7
2.1 INTRODUCCIÓN	7
2.2 DEFINICIÓN, PRINCIPALES CARACTERÍSTICAS Y BREVE DESCRIPCIÓN DE LAS PARTES:	8
2.3 REQUISITOS DE INSTALACIÓN	9
2.4 OBJETIVOS.....	9
2.5 PRINCIPALES CONCEPTOS	10
2.6 GENOTIPOS.....	12
2.6.1 DEFINICIÓN Y PRINCIPALES CARACTERÍSTICAS.....	12
2.6.2 MÓDULOS DEL GENOTIPO: DEFINICIÓN Y PROPIEDADES	15
2.6.3 CODIFICACIÓN DE GENOTIPOS	17
2.6.4 FORMATO F0.....	20
2.6.5 SINTAXIS.....	21
2.6.6 GENÉTICA	22
2.7 ALGORITMOS GENÉTICOS	23
2.7.1 DEFINICIÓN.....	23
2.7.2 SÍMIL BIOLÓGICO.....	24
2.7.3 ALGORITMOS GENÉTICOS APLICADOS EN FRAMSTICKS	26
2.7.4 ALGORITMOS GENÉTICOS DE SELECCIÓN	26
2.7.5 ALGORITMOS GENÉTICOS DE REEMPLAZO (ELIMINACIÓN DE INDIVIDUOS).....	28
2.7.6 OPERADORES GENÉTICOS (OGs)	28
2.7.6.1 Cruce	28
2.7.6.2 Mutación.....	31
2.7.6.3 Copia	32

2.7.7	ALGORITMOS GENÉTICOS DE EVALUACIÓN	32
2.7.8	LIMITACIONES DE LOS ALGORITMOS GENÉTICOS	33
2.8	SIMULACIÓN ARTIFICIAL EVOLUTIVA	35
2.8.1	TIPOS DE SIMULACIONES	36
2.8.2	ESTRUCTURA DE LA EVOLUCIÓN	37

CAPÍTULO 3 **41**

FRAMSTICKS GUI **41**

3.1	DEFINICIÓN Y CONCEPTOS	41
3.2	MÓDULOS QUE FORMAN FRAMSTICKS GUI.....	42
3.2.1	GUI COMO CREADOR Y EDITOR DE GENOTIPOS	42
3.2.1.1	Partes del editor	44
3.2.2	GUI COMO SIMULADOR DE PROCESOS EVOLUTIVOS	45
3.2.2.1	Conocimientos previos	46
3.2.2.2	Fases de la simulación	46
3.2.2.3	Secciones de la interfaz de simulación	47
3.2.2.4	Parámetros de la simulación	49
3.2.2.5	Genética en la evolución.....	50
3.3	SIMULACIONES LANZADAS DESDE LA TERMINAL, FRAMSTICKS CLI: COMMAND LINE INTERFACE.....	51
3.4	PASOS PARA REALIZAR UNA SIMULACIÓN	51

CAPÍTULO 4 **53**

FRED: FRAMSTICKS EDITOR..... **53**

4.1	DEFINICIÓN Y CARACTERÍSTICAS	53
4.2	ELEMENTOS DE LA INTERFAZ	54
4.2.1	MENÚ PRINCIPAL	54
4.2.2	VENTANA <i>BODY</i>	55
4.2.3	VENTANA <i>BRAIN</i>	57
4.2.4	VENTANA <i>MODEL PROPERTIES</i>	58
4.2.5	CONSOLA DE MENSAJES	59

CAPÍTULO 5 **60**

DISEÑO Y SIMULACIÓN DEL MODELO **60**

5.1	INTRODUCCIÓN.....	60
5.2	DISEÑO DEL MODELO	61
5.2.1	MODELO CON CUATRO EXTREMIDADES	62
5.2.2	MODELO CON CUATRO EXTREMIDADES ARÁCNIDO	64
5.2.3	PRUEBAS	66
5.2.4	ASPECTOS IMPORTANTES DE LAS SIMULACIONES	66
5.2.5	EVOLUCIÓN NEURONAL	68
5.2.6	EVOLUCIÓN MORFOLÓGICA	69
5.2.7	EVOLUCIÓN COMPLETA, MORFOLÓGICA Y NEURONAL.....	69
5.3	RESULTADOS.....	69
5.3.1	SIMULACIONES EVOLUTIVAS NEURONALES	70

5.3.1.1	Estudio previo de resultados del experimento 1	70
5.3.1.2	Resultado de pruebas de evolución neuronal	70
5.3.2	SIMULACIONES EVOLUTIVAS MORFOLÓGICAS	73
5.3.2.1	Pruebas de evolución morfológica	73
5.3.2.2	Estudio previo de resultados del experimento 2	73
5.3.2.3	Simulaciones relacionadas con la capacidad de la población.....	73
5.3.2.4	Simulaciones que afectan a los mecanismos de generación de descendencia.....	77
5.3.2.5	Resultados finales del experimento de evolución morfológica	79
5.3.3	SIMULACIONES EVOLUTIVAS COMPLETAS.....	80
5.3.3.1	Estudio previo de resultados de la evolución completa.....	80
5.3.3.2	Resultado de pruebas de evolución completa	81
5.3.3.3	Análisis del Ajuste en relación a la generación de genotipos	85
5.4	CONCLUSIONES DE LOS EXPERIMENTOS	86
5.5	ARTIFICIAL LIFE	89
5.5.1	INTRODUCCIÓN	89
5.5.2	DEFINICIÓN Y PRINCIPAL USO.....	89

MARCO SOCIO-ECONÓMICO92

6.1	PLANIFICACIÓN.....	92
6.2	PRESUPUESTO.....	94

CAPÍTULO 796

7.1	CONCLUSIONES	96
7.2	TRABAJOS FUTUROS	97

GLOSARIO98

REFERENCIAS98

ANEXOS100

MODELOS.....	100
A. FLINK	100
A.1. Código.....	100
A.2. Plano esquemático	101
A.3. Longitud de los nexos	102
B. IKKI.....	102
B.1. Código	102
C. IKKI'15	103
C.1. Código.....	103
C.2. Plano esquemático	105
C.3. Longitud de los nexos	105
D. IGOR''5	106
D.1. Código.....	106
D.2. Plano esquemático	107
D.3. Longitud de los nexos	108

ÍNDICE DE FIGURAS

Figura 2.5-1 Esquema de las partes que forman una simulación.....	11
Figura 2.6-1 Representación de un genotipo	13
Figura 2.6-2 Esquema de los tipos de Cromosomas	13
Figura 2.6-3 Esquema de las partes de un Cromosoma	13
Figura 2.6-4 Red neuronal como sistema de control del genotipo	14
Figura 2.6-5 Ejemplo de red neuronal	16
Figura 2.6-6 Esquema de una neurona	17
Figura 2.6-7 Mutación con cadena de caracteres estática.....	18
Figura 2.6-8 Mutación con cadena de caracteres dinámica	18
Figura 2.6-9 Cruce de genotipos.....	18
Figura 2.6-10 Ejemplo de genotipo “Articulación-Unión-Articulación”	19
Figura 2.6-11 Ejemplo de genotipo “Articulación-Unión-Articulación” mutado	19
Figura 2.6-12 Ejemplo de músculo con los ejes de rotación cambiados	19
Figura 2.7-1 Ejemplo de región del espacio	23
Figura 2.7-2 Esquema de las partes de un genotipo	25
Figura 2.7-3 Estructura de un cromosoma.....	25
Figura 2.7-4 Ejemplo de Proporción de Ajuste (Ruleta)	27
Figura 2.7-5 Cruce de cromosomas	28
Figura 2.7-6 Cruce de un punto	29
Figura 2.7-7 Cruce de dos puntos	29
Figura 2.7-8 Ejemplo de cruce uniforme	30
Figura 2.7-9 Ejemplo de estructura de un cromosoma	31
Figura 2.7-10 Ejemplo de algoritmo de Cruce	32
Figura 2.7-11 Esquema de la disposición de las soluciones y las regiones exploradas.....	34
Figura 2.7-12 Ejemplo ilustrado	35
Figura 2.8-1 Ejemplo de Flujo de una simulación evolutiva.....	36
Figura 2.8-2 Simulación evolutiva directa.....	37
Figura 2.8-3 Ejemplo traducido del libro de J, Koza [6]	39
Figura 2.8-4 Ejemplo pseudocódigo	40
Figura 3.2-1 Ejes de Coordenadas Framsticks GUI	43
Figura 3.2-2 Cambio de ejes de la aplicación FRED a Framsticks GUI	44
Figura 3.2-3 Editor de Framsticks GUI	44
Figura 3.2-4 Fases de la simulación.....	47
Figura 3.2-5 Secciones de la simulación	48
Figura 3.2-6 Ejemplos de los modelos analizados.....	48
Figura 3.2-7 Botones de la GUI.....	49
Figura 3.2-8 Parámetros de Simulación de la GUI.....	49
Figura 3.2-9 Capturas de pantalla de simulación en movimiento.....	50
Figura 3.3-1 Captura de pantalla de Framsticks CLI.....	51
Figura 4.1-1 Fred Editor	53
Figura 4.2-1 Botones del menú principal.....	54
Figura 4.2-2 Ejemplo de varios genotipos representados en la ventana <i>Body</i>	55
Figura 4.2-3 Vista de la ventana <i>Body</i>	55
Figura 4.2-4 Ejes de coordenadas en FRED	56

Figura 4.2-5 Botones de <i>Body</i>	56
Figura 4.2-6 Ventana de propiedades	57
Figura 4.2-7 Ventana <i>Brain</i> con un genotipo de ejemplo	57
Figura 4.2-8 Botones de <i>Brain</i>	58
Figura 4.2-9 Model properties	58
Figura 4.2-10 Consola de mensajes	59
Figura 5.1-1 Clasificación de las pruebas	60
Figura 5.2-1 Esquema de FLINK representado en Framsticks GUI.....	62
Figura 5.2-2 Planta de FLINK	63
Figura 5.2-3 Alzado de FLINK.....	63
Figura 5.2-4 Cerebro de FLINK	64
Figura 5.2-5 Modelo estilo araña versión 1	65
Figura 5.2-6 Modelo estilo araña versión 2	65
Figura 5.2-7 Caja negra.....	68
Figura 5.3-1 Cerebro de los modelos evolucionados.....	72
Figura 5.3-2 Resultados experimento 4 a, 4 b y 4 c.....	77
Figura 5.3-3 Esquema de la estructura de IKKI'8 obtenida del programa Framsticks GUI.	79
Figura 5.3-4 Gráfica de Ajuste vs Generación.....	81
Figura 5.3-5 Vistas de la estructura del genotipo IGOR''5	84
Figura 5.3-6 Cerebro de IGOR''5.....	84
Figura 5.3-7 Análisis Evolución completa.....	85
Figura 5.3-8 Vistas de IKKI'15	88
Figura 5.3-9 Planta modelo evolución natural	89
Figura 5.3-10 Alzado modelo evolución natural	89
Figura 5.5-1 Pantalla de Inicio Artificial Life	90
Figura 5.5-2 Simulación en 3D en Artificial Life.....	90
Figura 6.1-1 Diagrama de Gantt	93

ÍNDICE DE TABLAS

Tabla 2.6-1 Tipos de Funciones de estado de las neuronas	17
Tabla 2.6-2 Ejemplo Matriz de codificación f0	20
Tabla 2.6-3 Sintaxis f0	21
Tabla 2.7-1 Ejemplo de cruce de media	30
Tabla 2.7-2 Ejemplo de cruce geométrico	31
Tabla 2.7-3 Mutación	31
Tabla 2.7-4 Ejemplo de mutación con permutación del gen A	31
Tabla 5.2-1 Relación de parámetros configurados	68
Tabla 5.3-1 Valores mínimos	70
Tabla 5.3-2 Relación de experimentos de evolución neuronal	71
Tabla 5.3-3 Resultados experimentos evolución cerebral	71
Tabla 5.3-4 Valores mínimos del experimento 2	73
Tabla 5.3-5 Pruebas de evolución morfológica, parte 1	74
Tabla 5.3-6 Resultados simulación 4 a	75
Tabla 5.3-7 Resultados simulación 4 b	75
Tabla 5.3-8 Resultados simulación 4 c	76
Tabla 5.3-9 Pruebas de influencia de OGs, Experimentos 5 a y 5 b.	78
Tabla 5.3-10 Modelos evolucionados en las pruebas 5 a y 5 b	78
Tabla 5.3-11 Resumen de parámetros de las pruebas de evolución completa	82
Tabla 5.3-12 Resultado Simulación EXP'1	83
Tabla 5.3-13 Resultado Simulación EXP'2	83
Tabla 5.3-14 Resultado de la simulación de la prueba EXP'3	83
Tabla 5.3-15 Resultado mejores genotipos	87
Tabla 6.1-1 Planificación del trabajo	92
Tabla 6.2-1 Coste Personal	94
Tabla 6.2-2 Costes de Software y Hardware	95
Tabla 6.2-3 Costes totales	95
Tabla Anexo-1 Medidas de FLINK	102
Tabla Anexo-2 Medidas IKKI'15	105
Tabla Anexo-3 Medidas de IGOR''5	108

ÍNDICE DE ECUACIONES

Ecuación 2.6-1 Ecuación de Funcionamiento de una neurona artificial.....	16
Ecuación 2.7-1 Media Geométrica	30
Ecuación 2.7-2 Mutación de genotipos.....	32
Ecuación 2.7-3 Función de evaluación	33
Ecuación 2.7-4 Ajuste final.....	33
Ecuación 5.3-1 Ecuación de Ajuste	70
Ecuación 5.3-2 Ecuación de incremento de Ajuste	81
Ecuación 6.2-1 Fórmula de Costes	94

AGRADECIMIENTOS

Agradezco a toda mi familia el apoyo incondicional recibido y en especial a mi madre por haber tenido una paciencia desmesurada conmigo.

A Jorge Muñoz por haberme acompañado todo el verano haciendo arañas bailarinas, meditadoras y robots “insecto-palos”.

Por último a Tamara y Concha, por haberme ayudado durante todo el trabajo y haber sido fuente de inspiración y motivación.

RESUMEN

La finalidad de este proyecto es la documentación de la herramienta Framsticks y el desarrollo de un ensayo que cumpla un conjunto de objetivos definidos previamente.

Estos objetivos requieren el diseño de un modelo robótico experimental que emule el movimiento de desplazamiento animal; mediante evoluciones físicas y neuronales, se quiere modificar su estructura física para conseguir una moción rápida y natural.

Se deben desarrollar varios modelos, eligiendo el que mejor imite el movimiento animal y realizar distintas pruebas aplicando diversas técnicas genéticas hasta encontrar el modelo robótico evolutivo idóneo.

ABSTRACT

The aim of this project is the knowledge and research of Framsticks tool and development of a robotic design that fulfils some requirements defined previously.

This goal requires the implementation of an experimental robotic model that imitates animal performance; in order to approach this goal, it is necessary to generate some experiments that evolutionate its physical and neuronal structure on the way to improve its qualities to walk faster.

It is necessary to develop a set of models and choose the one who better imitates animal movements. It is also needed to create a group of tests applying different genetic methods until find the best robot that fits better.

Capítulo 1

Introducción y objetivos

1.1 *Introducción*

En sus orígenes, la robótica tenía como principal uso el desarrollo de equipos que facilitasen la vida a los humanos, desde robots “esclavos” hasta máquinas que pudieran realizar tareas en la industria, que normalmente eran desempeñadas por operarios. Estos modelos poseían un diseño fijo y sólo realizaban tareas programadas por sus creadores¹.

El inconveniente de estas máquinas aparecía cuando eran introducidas en entornos dinámicos dónde tenían que modificar y evolucionar sus tareas para adaptarse a las necesidades del entorno.

En este punto surgió la necesidad de robots autónomos que interaccionaran con un entorno dinámico, modificando sus funciones y desarrollando habilidades por sí solos, es decir, dotar de autonomía y capacidad de evolución a máquinas programadas.

A través de la robótica evolutiva se desarrollan modelos que proponen una solución adecuada a los problemas que se puedan encontrar las máquinas en un entorno dinámico; haciendo uso de distintas técnicas (Estrategias de evolución, programación genética y algoritmos genéticos) para encontrar la solución que mejor se adapte al conflicto. Mediante el empleo de estas técnicas se diseñan nuevos modelos que modifican el sistema de control del robot y su hardware para que desarrollen capacidades y aptitudes que los ayuden a adaptarse y realizar determinadas funciones en entornos dinámicos.

El principal inconveniente de la robótica evolutiva es el lento avance del proceso evolutivo sobre robots reales y su alto coste. La solución a este problema viene de la mano de la simulación artificial; el desarrollo de distintos programas que estudian y analizan el comportamiento de los robots y sus modelos evolucionados en entornos artificiales dinámicos.

Gracias a estas herramientas se pueden diseñar y probar soluciones previas a la construcción del modelo físico; reduciendo el tiempo de desarrollo y costes en hardware e impulsando el desarrollo de la robótica evolutiva. Debido a su flexibilidad y versatilidad permiten la creación y configuración de distintos modelos cuyo coste de fabricación sería muy elevado así como la realización de pruebas en multitud de escenarios sin poner en riesgo el prototipo.

Los primeros ensayos en este campo fueron realizados por Karl Sims [1], creador de un simulador de vida artificial de robots evolutivos. Este programa está basado en la evolución de distintos modelos robóticos en base a su adaptación al medio, a través de la alteración de sus

¹Por ejemplo el robot Unimate, que fue uno de los primeros robots industriales que se desarrollaron y su principal uso era el transporte de piezas en la cadena de montaje de General Motors.

características; como es una herramienta de código cerrado no se pueden modificar los modelos ni parámetros de la evolución.

Otros simuladores como Webots permiten programar, modelar y simular robots móviles; pudiendo elegir las propiedades de los objetos, la forma, el color, los sensores y actuadores de cada modelo y simularlo en un entorno 3D. Sin embargo los prototipos no pueden evolucionar ni modificar sus propiedades por sí solos, reduciendo sus funciones, utilidad y versatilidad. [2]

Framsticks propone una herramienta multiplataforma de código abierto basada en el simulador de Sims; donde el usuario puede crear sus propios modelos y configurar las técnicas genéticas que actúan sobre ellos. Gracias a Framsticks, se pueden evolucionar distintos modelos robóticos y analizar como se desenvuelven en el entorno artificial configurando todos los parámetros que participan.

Como la intención es evolucionar y documentar un modelo robótico animal para aumentar su velocidad de desplazamiento, se realiza el estudio de esta herramienta y se elaboran distintas pruebas para encontrar la solución que mejor se adapte. El prototipo deberá imitar² el movimiento animal; por tanto, su versión evolucionada tendrá que caminar más rápido pero sin perder su carácter “natural”.

1.2 Objetivos

El objetivo fundamental del trabajo es el estudio de la herramienta, diseño y análisis de la evolución de un modelo robótico implementado en ella. El modelo configurado deberá emular, en la medida de lo posible, el desplazamiento animal.

La evolución tendrá lugar principalmente en su morfología, produciendo prototipos más rápidos que conserven la estructura física inicial³ pero modificando la longitud y orientación de sus extremidades.

En base a este objetivo principal, se proponen los siguientes objetivos parciales:

- Documentación de la herramienta: Estudio de las funciones que pueden realizarse en cada una de las plataformas y el papel que desempeña cada una en el experimento robótico evolutivo.
- Diseño del modelo: Creación de un modelo robótico que imite el movimiento animal mediante la configuración de su morfología y sistema de control.
- Realización de ensayos evolutivos: Desarrollo de un conjunto de pruebas, utilizando el modelo diseñado; con el objetivo de incrementar la velocidad y distancia recorrida por el individuo a través de la evolución de sus características físicas y neuronales.
- Validación y análisis de resultados: Análisis crítico de los modelos evolucionados que han sido generados en las distintas pruebas; selección de los robots atendiendo a criterios establecidos previamente.
- Conclusiones del experimento: Estudio de la influencia de las técnicas evolutivas empleadas en los ensayos evolutivos y análisis de los resultados obtenidos.

² El cumplimiento de este objetivo depende de las características y restricciones del programa.

³ Los nuevos prototipos deben conservar tanto el número de articulaciones como de extremidades del modelo inicial.

1.3 Estructura de la memoria

Para facilitar la lectura de este trabajo, a continuación se hace una breve descripción de la distribución de la memoria:

- **CAPÍTULO 1 “INTRODUCCIÓN”:** En este capítulo se expone el contexto en el que se realiza el trabajo así como un breve resumen del contenido y objetivos principales de la memoria.
- **CAPÍTULO 2 “FRAMSTICKS”:** Descripción de la herramienta utilizada, incluyendo las características de los elementos que la componen. Se centra en la explicación de los principales aspectos de la herramienta.
- **CAPÍTULO 3 “FRAMSTICKS GUP”:** En este capítulo se detalla el software de simulación empleado para el desarrollo de robots evolutivos y sus funcionalidades específicas.
- **CAPÍTULO 4 “FRED: FRAMSTICKS EDITOR”:** Describe los aspectos más importantes de la aplicación FRED y sus distintas partes.
- **CAPÍTULO 5 “DISEÑO Y SIMULACIÓN DEL MODELO”:** Se describen los modelos propuestos para el experimento y los ensayos evolutivos realizados. Se incluye también los resultados y las conclusiones derivadas de los mismos.
- **CAPÍTULO 6 “MARCO SOCIOECONÓMICO”:** Incluye la planificación y presupuesto detallado del proyecto.
- **CAPÍTULO 7 “CONCLUSIONES Y TRABAJOS FUTUROS”:** Análisis crítico de los resultados del trabajo y se comentan las posibles propuestas de mejora.
- **CAPÍTULO 8 “GLOSARIO Y REFERENCIAS”:** Se incluyen las fuentes de información empleadas para la realización del proyecto y un breve catálogo de las abreviaturas utilizadas.
- **CAPÍTULO 9 “ANEXOS”:** En este último capítulo de la memoria se encuentran los planos esquemáticos y código del robot inicial y de los mejores modelos obtenidos en los ensayos evolutivos.

CAPÍTULO 2

FRAMSTICKS

2.1 *Introducción*

Framsticks surgió como un simulador de vida artificial, aunque su alcance es mayor gracias al desarrollo de modelos virtuales que emulan y evolucionan el comportamiento de diseños robóticos. Engloba todas las funciones necesarias para la ejecución de un experimento, desde la creación del modelo, configuración de sus distintas partes, simulación en diferentes entornos y su evolución en otros robots.

Se asienta sobre tres pilares fundamentales, la simulación mecánica en 3D del comportamiento de los genotipos en un entorno artificial, la actuación del sistema de control sobre los individuos y su evolución hacia un objetivo definido por el usuario.

En relación con los dos primeros pilares, permite la creación, diseño y edición de modelos robóticos que integren un sistema de control (red neuronal) y una estructura física (morfología) en un mismo ente; incluyendo el análisis y simulación virtual de su comportamiento en un entorno artificial para poder estudiar sus movimientos e interacciones con el medio.

El último pilar y más importante, es la evolución de prototipos con la intención de desarrollar y optimizar determinadas habilidades o características de los modelos. Para ello se sirve de los algoritmos genéticos, como herramienta fundamental para desempeñar esta tarea

Actualmente existen varios editores y simuladores de modelos robóticos, por ejemplo Webots⁴, cuyas funciones son similares a Framsticks, simulan robots en entornos artificiales configurables y permiten la edición de las partes que forman al modelo. Sin embargo esta herramienta no tiene implementada la evolución de los modelos; impidiendo el desarrollo evolutivo de los prototipos así como la optimización de sus funciones a través de estrategias de evolución.

Por el contrario, el simulador de Karl Sims, sí que tiene incluida esta funcionalidad pero no se pueden modificar los modelos robóticos. Por tanto se considera que Framsticks es la herramienta más completa para la simulación de robots evolutivos.

⁴ Para más información sobre la herramienta Webots consultar[2]

2.2 Definición, principales características y breve descripción de las partes:

Es un programa creador y editor de prototipos, desde su estructura física hasta la distribución de sus neuronas; permite la selección de las características de la simulación, la capacidad de la población, los operadores genéticos y el resto de operadores genéticos que participan en la evolución; tiene un historial evolutivo que guarda todos los cambios que sufre cada una de las generaciones y permite la personalización del objetivo de la simulación.

Constituye una herramienta completa para el análisis y creación de experimentos relacionados con la robótica evolutiva; es versátil, posee distintas versiones para varios sistemas operativos. Al tratarse de una plataforma de desarrollo de software libre admite modificaciones en el programa y la incorporación de nuevas variables, propiedades, tipos de evoluciones y simulaciones diseñadas por el propio usuario de la herramienta.

Esta plataforma trabaja con un tipo de archivos propios; para los genotipos los archivos son .gen y las simulaciones son .expef; todos estos archivos son editables y se pueden abrir con programas de escritura.

Framsticks permite al usuario realizar un experimento completo, desde el diseño de su estructura morfológica y neuronal, hasta el desarrollo de sus habilidades por medio de evoluciones, espontáneas o predeterminadas, en entornos de simulación por ordenador.

Como se ha explicado en el apartado anterior, los tres pilares sobre los que se asienta Framsticks son los encargados de llevar a cabo todo el proceso evolutivo del experimento. Las diferentes funciones se pueden ejecutar en los distintos módulos del programa. A continuación se describe cada uno de los módulos y las tareas que pueden desempeñar.

La plataforma está formada por un conjunto de aplicaciones que permiten la realización del estudio evolutivo completo, son las siguientes:

- *Framsticks Guide User Interface* (GUI): Programa encargado de la creación, evolución y representación virtual de genotipos en un entorno artificial. A través de ella se crean, editan y evolucionan robots virtualmente.
- *Framsticks Editor* (FRED): Diseñador de prototipos, permite la creación de modelos a partir de la unión de articulaciones, nexos e implementación del cerebro. Se encarga de configurar un conjunto de variables relacionadas con la morfología y el sistema de control de los robots.
- *Artificial Life* (AL): Simulador gráfico en 3D de vida artificial. Los usuarios pueden observar el comportamiento de distintos experimentos y su evolución, todo ello en un entorno virtual en 3D que también puede ser configurado.
- *Framsticks Comand Line Interface* (CLI): Interfaz de línea de comandos encargada de la ejecución y configuración de simulaciones a través de un conjunto de comandos específicos lanzados desde la terminal. Dirigida a usuarios avanzados ya que es necesario el conocimiento de los comandos de ejecución. Es más rápida que la GUI al prescindir de las simulaciones gráficas del entorno.

2.3 Requisitos de instalación

Para poder instalar la herramienta es necesario tener determinadas especificaciones dependiendo del sistema operativo (SO) del ordenador.

En el caso de Windows para el uso de Framsticks GUI y Artificial Life, los archivos disponibles en la página web son de tipo .exe, por lo que se pueden ejecutar en cualquier entorno de Windows. Sin embargo, si se quiere utilizar FRED, se debe instalar un entorno de desarrollador de programas en Java.

Si su SO es Linux, se necesita una re implementación de la interfaz de programación de aplicaciones de Win 32 para SO basados en Unix, que permita la ejecución de programas basados en MS-DOS y Windows, para poder utilizar Framsticks GUI y Artificial Life. En el caso de que se quiera utilizar FRED, es también recomendable instalar un entorno desarrollador de programas en Java, como por ejemplo Open java.

2.4 Objetivos

Esta herramienta es un programa basado en la computación evolutiva, la inteligencia artificial y el estudio de redes neuronales de un conjunto o población de modelos robóticos; creados por el propio usuario en un entorno de simulación virtual.

Permite el estudio evolutivo de un único individuo o conjunto a través de simulaciones⁵ que generan modelos evolucionados de los prototipos iniciales. Mediante estas simulaciones se analiza el comportamiento, interacción y evolución de cada uno de los individuos con respecto al resto. También se pueden crear sus propios modelos, pudiendo definir desde la estructura física del individuo hasta el diseño del cerebro.

Gracias a este programa el usuario puede participar en todo el experimento, desde el diseño de los prototipos, la simulación en entornos artificiales del comportamiento y evolución de los sujetos y el análisis de los datos obtenidos y las distintas técnicas evolutivas empleadas.

Como se explica anteriormente, uno de los primeros programas de simulaciones de vida artificial, fue el creado por Karl Sims, estas simulaciones están completamente cerradas, no es posible saber qué parámetros afectan a cada una de ellas además el diseño de la morfología y sistema de control de los genotipos no es posible realizarla, ya que los modelos son cargados directamente sobre la plataforma. A diferencia de Sims, Framsticks, al ser de código abierto, supone un valor añadido comparado con otras herramientas, ya que el usuario puede configurar todos los parámetros que participan en los experimentos.

⁵ Las simulaciones se guían por un conjunto de instrucciones basadas en AGs y otras técnicas evolutivas. Este conjunto de instrucciones busca la solución más óptima de un determinado problema mediante la explotación de todas las regiones del espacio disponibles.

2.5 Principales conceptos

Para un primer contacto con la herramienta es necesario definir los conceptos básicos con los que opera, a continuación se hace una breve descripción de los elementos que forman parte del programa:

- **Genotipo:** Se denomina a todo modelo que forme parte del proceso evolutivo, interactúe y evolucione en el entorno artificial creado por Framsticks. Este término no sólo hace referencia a los individuos diseñados por la plataforma, sino también a los modelos creados por el propio usuario a través del editor de genotipos. A diferencia de la biología, donde se hace distinción entre genotipo y fenotipo, en esta herramienta no se tiene en cuenta esta diferenciación⁶. Los genotipos representan tanto la estructura genética del individuo como su representación en el entorno artificial, todas las instancias (fenotipos) de un mismo genotipo poseen las mismas características.
- **Grupos:** Son las distintas concentraciones en las que se dividen los genotipos que participan en las simulaciones; distinguiéndose dos clases, el conjunto de genotipos y la población temporal. Dentro de la población temporal, se dividen en dos tipos, los individuos simulados como criaturas y los que forman parte de la comida de las criaturas. En la Figura 2.5-1, se muestran los distintos grupos de genotipos que pueden participar en las simulaciones.
- **Conjunto de genotipos:** Hace referencia a la lista⁷ de individuos de una población que poseen diferente estructura genética o pertenecen a distinta generación. Se almacenan todos los sujetos que pueden formar parte de la simulación; sobre este conjunto se aplican los AGs.
- **Población temporal:** Agrupación de genotipos simulados en un momento determinado. Hay dos grupos distintos: Las criaturas, descendientes del conjunto de genotipos sobre los que se ha aplicado los Operadores Genéticos (OGs) que han provocado la reproducción y transformación de los individuos de una misma sociedad. La comida es el segundo grupo de genotipos de la población, es la fuente de energía de las criaturas⁸. El objetivo de la población es analizar el comportamiento de los genotipos que luego serán seleccionados para continuar la especie. Se pueden generar distintas instancias de un mismo genotipo para actualizar la información de la simulación.
- **Criaturas:** Son los distintos genotipos que forman parte de la población. Cada criatura puede ser representada en el entorno artificial a través de las instancias creadas a partir de los genotipos de la simulación.
- **Comida:** Fuente de alimentación y energía de las criaturas. Pueden ser agrupadas en la población temporal e interactúan con los genotipos simulados pero no evolucionan.
- **Genética:** Aspecto determinante en la evolución de los genotipos. Permite la inalteración de las propiedades de determinadas variables durante la evolución, pudiendo por ejemplo fijar el número y el tipo de neuronas que conforman la red neural de los prototipos. La genética es el conjunto de variables y características que posee un individuo que pueden ser modificadas o

⁶ En el campo de la Biología, el genotipo guarda toda la información genética del individuo mientras que el fenotipo es la instanciación de esta información. En el fenotipo se reproducen los caracteres genéticos guardados en el genotipo, pero no todos, provocando diferencias entre ambos [3].

⁷ Esta lista incluye todas las características de cada uno de los individuos que intervienen en las simulaciones, así como el número de instancias que han sido creadas para cada genotipo.

⁸ Todos los miembros del conjunto de genotipos pueden ser simulados como criaturas o como comida, la diferencia radica en que las criaturas forman parte del proceso evolutivo y como tal interaccionan con el entorno, al contrario que los sujetos que se simulan como comida, que no interactúan y su única función es la de servir como fuente de energía para las criaturas.

inalteradas durante el proceso evolutivo⁹.

- **Algoritmos genéticos (AGs):** Herramienta matemática de la que se sirve la robótica evolutiva para cumplir un objetivo determinado. Conjunto de instrucciones que alteran las propiedades físicas y neuronales de los individuos a través de mecanismos que generan nuevos modelos evolucionados.
- **Simulación:** Representación virtual y análisis del comportamiento de los genotipos en un entorno artificial durante la ejecución de su evolución. En estas simulaciones se aplican las técnicas evolutivas que provocan la evolución de una determinada población de individuos. Framsticks permite la creación de multitud de simulaciones dependiendo de la meta del usuario. A través de ellas, se estudia el comportamiento de los sujetos, provocan procesos evolutivos que alteran sus propiedades y habilidades con el objetivo de mejorar y optimizar determinadas capacidades o aptitudes de los prototipos.
- **Parámetros de la simulación:** Conjunto de variables que gobiernan la simulación, algunos de ellos componen las premisas de los AGs. Activan los distintos AGs que han sido definidos en la simulación, provocando el nacimiento, evolución, reproducción, muerte y cambio generacional de todos los individuos que forman parte de la simulación. Se encargan también de controlar el avance de la simulación y el análisis de los genotipos.
- **Estado de la simulación:** Captura de un momento determinado de una simulación. Recopilación de todas las variables, sujetos, poblaciones y genéticas que se reproducen en un instante dado. Documento¹⁰ en el que se almacenan toda la información relativa a todos los parámetros de una simulación concreta en un momento determinado.

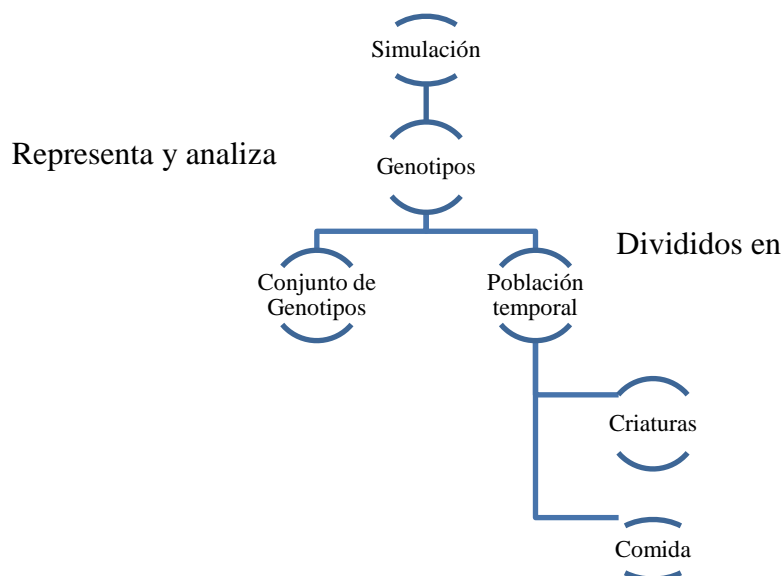


Figura 2.5-1 Esquema de las partes que forman una simulación

Los individuos agrupados en el conjunto de genotipos (Figura 2.5-1), son los modelos sobre los que se llevará a cabo la evolución. Se realiza un conjunto de instancias válidas de cada uno de ellos (población temporal), estas instancias son simuladas virtualmente en un entorno

⁹ Por ejemplo si en la genética está definido que el número de neuronas tiene que ser inferior a diez, cuando se ejecuten los distintos algoritmos genéticos (mutación, cruce etc), los genotipos cuyo cerebro posea más de diez neuronas no serán válidos y por lo tanto se eliminarán.

¹⁰ Gracias a este documento, se puede recopilar información relacionada con la genética y el ajuste de un determinado conjunto de genotipos para su posterior análisis.

artificial; dependiendo del grupo en el que estén, formarán parte de las criaturas de las que se analiza su comportamiento o pasarán a formar parte de la comida de las criaturas mencionadas anteriormente.

En el caso de que pertenezcan al grupo de las criaturas, se realizará un análisis de su actuación en el entorno y se les asignará una posición en el ranking del conjunto de genotipos (valor de Ajuste¹¹). Por el contrario, si forman parte del grupo de la comida, servirán de fuente de energía para los genotipos que están siendo evaluados.

Para poder evolucionar los genotipos, es necesario seguir una serie de instrucciones que guían la simulación y el desarrollo de la evolución. Los algoritmos genéticos¹² (AGs) son el conjunto de instrucciones encargados de este proceso, existen varios tipos dependiendo de la tarea que realicen.

Además hay un grupo de parámetros que determinan la evolución y que limitan la aplicación de los AGs, la genética; dependiendo del tipo de codificación, la genética fija las limitaciones y el tipo de evolución que se puede ejecutar.

Por último, en esta plataforma están incluidas un conjunto de variables que afectan a la monitorización, visualización y configuración de cada una de las simulaciones que influyen también en las simulaciones. Como pueden ser la configuración del entorno artificial, monitorización de las simulaciones etc.

2.6 Genotipos

2.6.1 Definición y principales características

Conjunto de sujetos que constituyen una posible solución a un determinado conflicto. Cada uno de ellos es un punto de exploración de una región del espacio de todas las posibles soluciones a un problema determinado. Son los modelos¹³, objetos de estudio de las simulaciones, se representan en un entorno computacional y se someten a distintos procesos de transformación. Están formados por dos módulos, el cuerpo y el cerebro que se relacionan a través de la red neuronal, entendiéndose como red neuronal el conjunto de los distintos tipos de neuronas que gobiernan el individuo y que están conectadas entre ellas; comprenden el sistema de control de individuo.

Un genotipo, posee unas características morfológicas (cuerpo) y neuronales (cerebro) que lo hacen único, en la Figura 2.6-1, se observa un ejemplo de la estructura física de un genotipo. Estos atributos son almacenados en sus cromosomas, variables que son alteradas a través de los distintos procesos evolutivos que se provocan mediante las simulaciones.

¹¹ Para más información sobre el Ranking del conjunto de genotipos así como su valor de ajuste consultar el apartado 2.7 donde se explica detalladamente.

¹² En el apartado 2.7 se explican detalladamente.

¹³ Para más información sobre los modelos de genotipos desarrollados en Framsticks consultar [16]

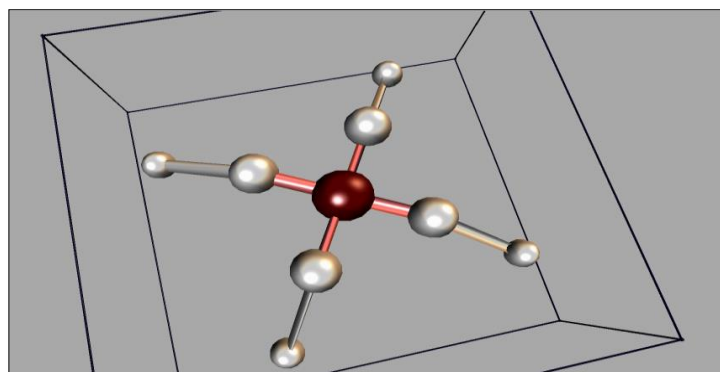


Figura 2.6-1 Representación de un genotipo¹⁴

Haciendo un símil biológico, cada cromosoma está formado por una serie de genes y estos genes y su disposición es lo que caracteriza a cada individuo. En términos relacionados con Framsticks, la equivalencia se explica en la Figura 2.6-2 y Figura 2.6-3.

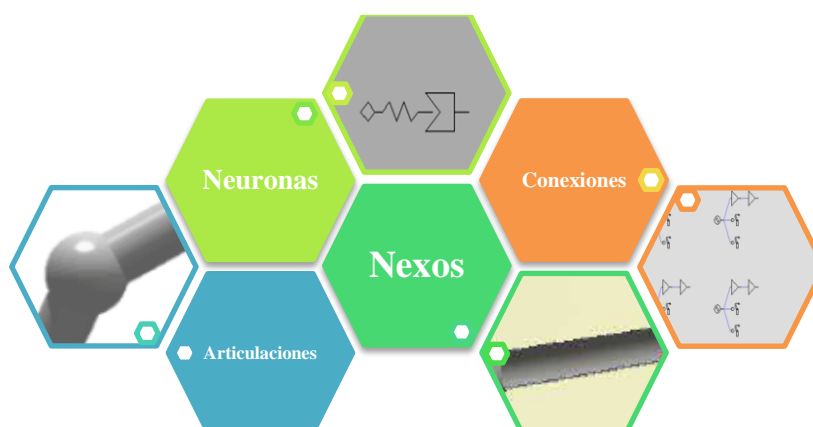


Figura 2.6-2 Esquema de los tipos de Cromosomas¹⁵

Donde cada genotipo está formado por un conjunto de cromosomas, que puede ser de cuatro tipos: neuronas, articulaciones, nexos y conexiones; cada tipo posee un grupo de características que son los distintos genes que lo forman. Es decir los distintos tipos de Cromosomas son las partes que conforman un genotipo mientras que las propiedades que caracterizan a cada una de estas partes, serían los genes.



Figura 2.6-3 Esquema de las partes de un Cromosoma

¹⁴ Capturade pantalla de un genotipo representado en FRED.

¹⁵ Las imágenes que aparecen en los hexágonos han sido obtenidas del Manual de Framsticks[4].

Los Cromosomas representarían todas las variables que caracterizan a un individuo, sus nexos, articulaciones, neuronas que gobiernan su cerebro del individuo y las conexiones entre ellas. Poseen unas propiedades que los caracterizan, estas propiedades serían los distintos genes que conforman un cromosoma.

Las propiedades de cada cromosoma son distintas, en el caso de las articulaciones y nexos, sus genes hacen referencia a parámetros como la masa, la fricción, la resistencia, dureza.

Por el contrario en las conexiones neuronales, estas propiedades se caracterizan por el peso que tienen sobre las decisiones que toma el individuo. En los cromosomas neuronales, son los distintos tipos de neuronas, que pueden aportar diferentes habilidades y aptitudes a los genotipos.

Las conexiones entre neuronas y la disposición de las partes o miembros de un individuo, son de vital importancia ya que influyen en el comportamiento del individuo y conforman la red neuronal del modelo, el sistema de control del genotipo (Figura 2.6-4).

Las redes neuronales artificiales intentan emular el comportamiento del cerebro humano ejecutando acciones propias del conocimiento de las personas, como aprender, memorizar, en resumen, controlan todos los movimientos y habilidades que pueden ejecutar y desarrollar un individuo.

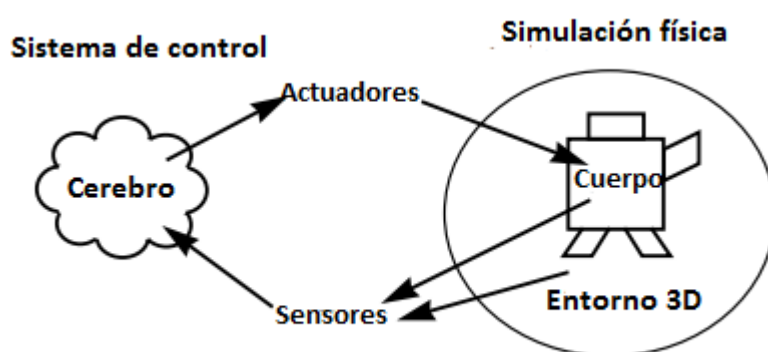


Figura 2.6-4 Red neuronal como sistema de control del genotipo¹⁶

Una red neuronal es un conjunto de elementos conectados entre ellos de una determinada forma. Estos elementos son las neuronas artificiales cuyo funcionamiento es similar al de un procesador, a partir de un conjunto de entradas genera una función de salida que depende de un estado u función de activación (caja negra) cuyos parámetros están ocultos. La potencia y complejidad de la red neuronal se mide por el número de interconexiones entre las distintas neuronas.

Para el desarrollo de habilidades, es necesario el aprendizaje. Este se consigue a través de reglas que adaptan el peso de las conexiones de las distintas neuronas para producir los movimientos o comportamientos específicos que desarrollan las habilidades del individuo.

Además de todos estos parámetros, se pueden definir otras características de los genotipos, más relacionadas con su comportamiento durante su período de vida, como por ejemplo,

¹⁶ Esta figura ha sido traducida de la publicación de Karl Sims [1].

aumentar o disminuir la velocidad con la que envejecen, alterar el aporte energético que le proporciona la comida.

2.6.2 Módulos del Genotipo: Definición y propiedades

Un genotipo está formado por dos partes, la estructura física y el sistema de control de dicha estructura; cada una de estas partes está formada por distintos módulos y conexiones entre ellos. Framsticks permite el diseño de cada individuo atendiendo a la implementación de ambos módulos: cuerpo (nexos, articulaciones) y cerebro (neuronas y conexiones) a continuación se hace una descripción de cada uno de los elementos y sus propiedades:

- **Articulación:** Elemento de unión entre dos nexos, según el modelo¹⁷ de Framsticks, el número máximo de uniones de una articulación es dos, por lo tanto cada una de ellas puede unirse como máximo a dos nexos. Las propiedades que las caracterizan son de dos tipos: propiedades geométricas y físicas relativas a su posición en el espacio y propiedades biológicas relacionadas con las capacidades de adaptación al medio de cada uno de los genotipos. En cuanto a sus propiedades geométricas, por defecto se sitúa el origen en la ubicación de la primera articulación creada (*Part0*¹⁸). Se definen atendiendo a su ubicación y rotación respecto del plano x, y, z, que dependiendo de la aplicación en la que se represente el genotipo varían. Los parámetros que definen su localización y rotación, son los siguientes: (x, y, z, rz, rx, ry). Dependiendo del tipo de lenguaje de codificación utilizado, estos parámetros están visibles o aparecen configurados por defecto en la sintaxis de codificación.
- **Nexos:** Son las extremidades del objeto, que se unen mediante articulaciones. El gen más importante que caracterizan a este cromosoma son las propiedades geométricas. Relacionadas con su rotación respecto de los planos de coordenadas, no poseen valores de posición respecto del origen ya que son el nexo de unión de dos partes y la forma de expresar su posición es mediante la diferencia de longitud entre las partes que los unen.
- **Neuronas:** Encargadas de gobernar el comportamiento de un prototipo, conforman el sistema de control del genotipo. Dependiendo de las funciones que realicen se pueden distinguir dos tipos de neuronas, neuronas-artificiales ó tipo sensor y neuronas tipo actuador. Además del tipo de neurona, existen otros dos parámetros que caracterizan a cada una de ellas y son sus conexiones y el peso o aporte de cada conexión.
 - Conexiones: Son todos los enlaces que puede poseer cada tipo de neurona. Dependiendo del tipo de neurona, estos enlaces pueden estar limitados, en [4] se explican los distintos casos.
 - Pesos: Es el aporte o grado de excitación de las señales de entrada, a las que se une una neurona, que alteran su señal de entrada, que unida al estado de la neurona, afecta a la salida.

Tal como señala el profesor Xavier Basogain [5] “*La neurona artificial fue diseñada para 'emular' las características del funcionamiento básico de la neurona biológica. Para poder 'emular' este comportamiento, cada neurona tiene que tener un 'peso' que determina su grado de aporte a las acciones que realiza un individuo y que determinan su conducta; además de su contribución, cada neurona se caracteriza por sus nexos de unión con el resto de neuronas. Por lo que una única neurona puede influir en el comportamiento del resto, por lo que cada una,*

¹⁷ Framsticks tiene limitado el número de articulaciones que se pueden unir a un nexo en dos, pero como es una plataforma de código abierto, el usuario puede modificar esta limitación aunque en este trabajo se mantienen.[4]

¹⁸ Nombre asignado por la herramienta a la primera articulación creada.

tiene que estar definida por tres parámetros, el tipo de neurona, el 'peso' que aporta a la actitud del individuo y las conexiones que establece.”

No sólo el tipo de neurona determina el comportamiento del individuo, sino también los pesos y las distintas conexiones entre ellas. La red neuronal juega un papel importante¹⁹ en el comportamiento de cada uno de los individuos, es el sistema de control de todos los movimientos y comportamientos de cada uno de los genotipos. Como se ha explicado en el apartado anterior (2.6.1), para que los genotipos desarrollen nuevas habilidades, es necesario el aprendizaje y para ello hay que configurar la red neuronal del genotipo.

Las neuronas ejecutan funciones parecidas a las de un procesador, transforman un conjunto de entradas, en función de su estado, en una salida determinada. Dependiendo del valor de salida, se trata de una neurona ‘excitadora’, en el caso de valores positivos, o ‘inhibidora’, si por el contrario el valor de salida es un número negativo.

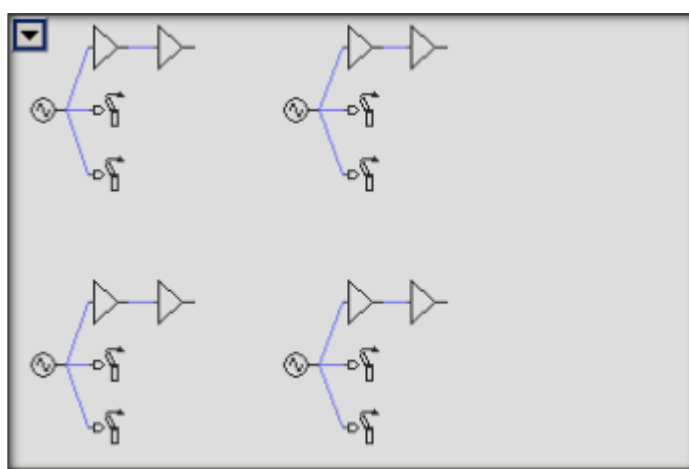


Figura 2.6-5 Ejemplo de red neuronal²⁰

Se puede comparar con una caja negra que recibe una serie de entradas y da una salida en función de una serie de parámetros internos de la neurona (estado), a continuación, en la Figura 2.6-6 y Ecuación 2.6-1, se muestra un esquema de como sería. Por otro lado, en la Figura 2.6 -5 se encuentra la representación de una red neuronal en la aplicación FRED, en este ejemplo, la red neuronal está dividida en cuatro secciones aisladas, formadas por dos neuronas actuadores, dos neuronas sigmoideas y una neurona senoidal que las gobierna.

$$\sum Entradas = Función\ de\ Estado * Peso \rightarrow Función\ de\ Salida$$

Ecuación 2.6-1 Ecuación de Funcionamiento de una neurona artificial

Como se ha comentado anteriormente, las neuronas se pueden clasificar como neuronas sensores, que recogen información del ambiente o del genotipo y la procesan produciendo una salida; ó por el contrario, neuronas actuadores, que se comportan como los músculos del individuo. Las neuronas actuadores ejecutan acciones ó generan una señal de salida en función de la señal que reciben.

¹⁹ Uno de los ensayos realizados en este trabajo tiene como objetivo evaluar la influencia de las conexiones y distintos pesos de las neuronas en el comportamiento y la evolución de un conjunto de individuos.

²⁰ Imagen obtenida del programa FRED, de Framsticks.

Sin embargo, las neuronas sensores utilizadas en este trabajo, procesan la señal de entrada y en función de dicha señal y el estado de la neurona, generan una salida.

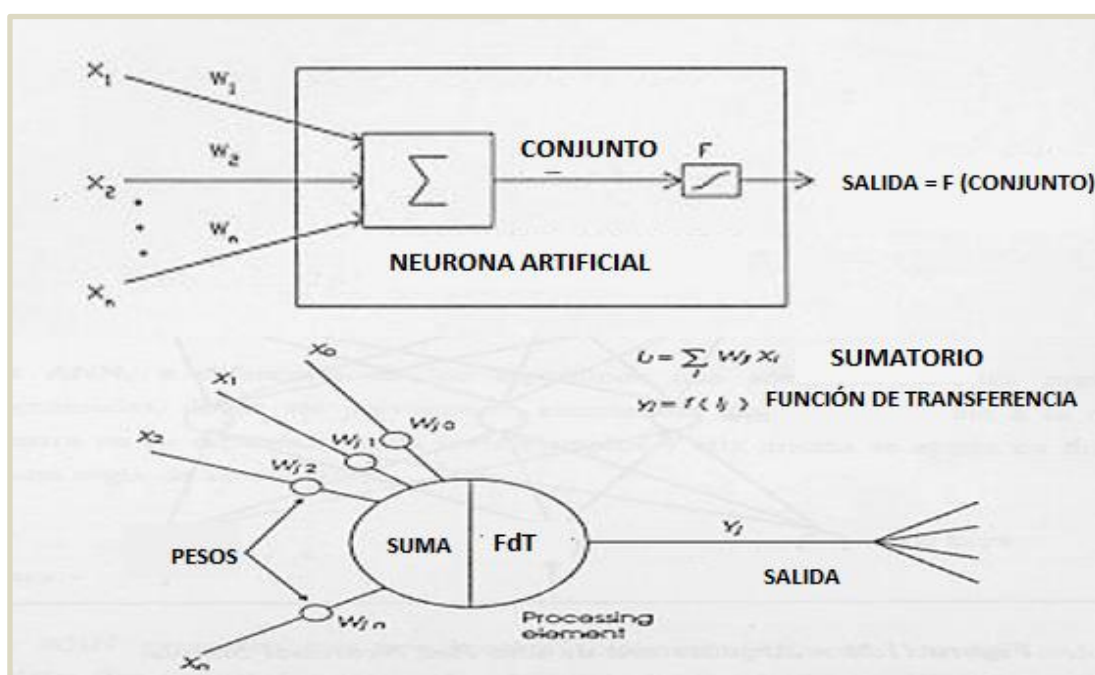


Figura 2.6-6 Esquema de una neurona ²¹

La función estado de una neurona artificial, comúnmente suele ser de dos tipos, Función Sigmoidea y la función Tangente hiperbólica, a continuación se muestran las funciones (Tabla 2.6-1). En el caso particular de Framsticks, se va a utilizar neuronas que utilicen la función Sigmoidea como la función que regula el estado de las neuronas artificiales.

Tabla 2.6-1 Tipos de Funciones de estado de las neuronas ²²

Función	Ecuación
Sigmoidea	$Salida = \frac{1}{1 + e^{-t}}$
Tangente Hiperbólica	$Salida = \tanh t$

2.6.3 Codificación de Genotipos

La representación de un individuo es una de las limitaciones más importantes del proceso evolutivo; la manera de expresar las propiedades y distintas variables de un genotipo condiciona su evolución, limitando algunas transformaciones [6].

Existen varias formas de expresar sus propiedades; mediante cadenas de caracteres de longitud fija o a través de lenguajes de programación que generen matrices que incluyan las distintas partes que forman el genotipo.

²¹ Esquema traducido del libro de Xabier Basogain [5].

²² Donde t es el estado en el que se encuentra la neurona

La elección del tipo de codificación de un individuo delimita sus posibilidades de evolución, por ejemplo, al usar cadenas de caracteres de longitud fija, las mutaciones de las variables se restringen a la longitud de la cadena, impidiendo o invalidando el nuevo genotipo, como se observa en la Figura 2.6-7, donde la variable mutada AA' no es válida ya que la longitud de la cadena es de tres caracteres, por lo que el genotipo B será una instancia fallida.

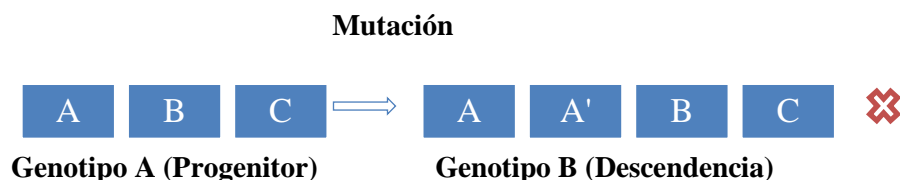


Figura 2.6-7 Mutación con cadena de caracteres estática

Sin embargo utilizando cadenas dinámicas o programas que generen matrices o cadenas de caracteres dinámicas, cuya longitud pueda variar sin ninguna restricción. Las limitaciones de evolución disminuyen, ya que permiten realizar más transformaciones sobre los individuos. Por ejemplo, en la Figura 2.6-8, se realiza la misma mutación que en el ejemplo anterior, pero utilizando una cadena de caracteres dinámica, el genotipo B es válido ya que la longitud de la cadena de caracteres puede variar, abriendo una línea de evolución que no sería posible utilizando codificaciones de cadenas de caracteres estáticas.

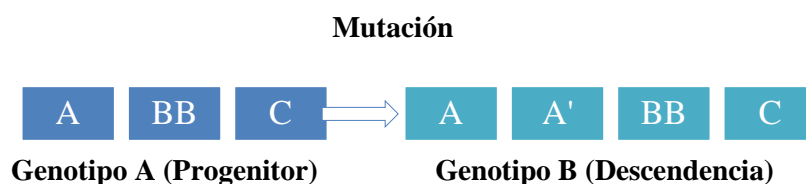


Figura 2.6-8 Mutación con cadena de caracteres dinámica

En el caso del cruce, sucede lo mismo, en la Figura 2.6-9, se cruzan dos genotipos, A y A' obteniendo el genotipo B, cuya longitud de caracteres es superior, pero al igual que en el caso anterior, el genotipo B es válido ya que la cadena de caracteres tiene una longitud variable.

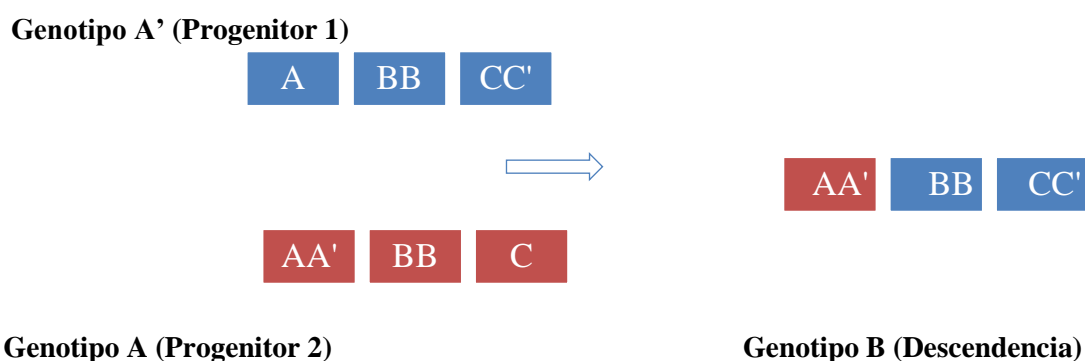


Figura 2.6-9 Cruce de genotipos

Son aplicables también al siguiente ejemplo, una matriz formada por un conjunto de cadenas de caracteres, donde cada fila es un módulo distinto del genotipo y cada columna son las

características de cada módulo, por ejemplo, las filas 1 y 3 son dos articulaciones, mientras que la fila 2 es la unión entre ambas.

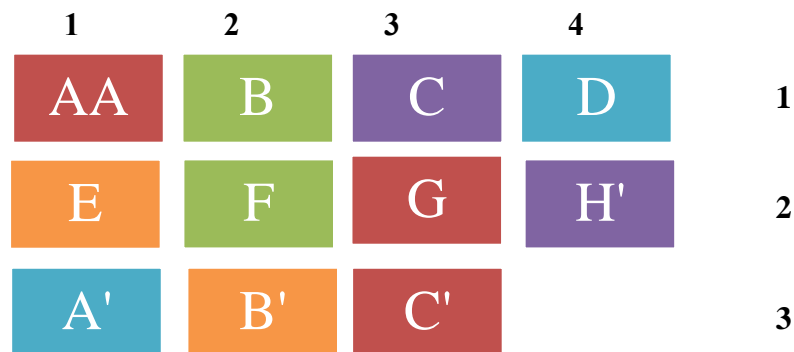


Figura 2.6-10 Ejemplo de genotipo “Articulación-Unión-Articulación”

Como puede se observa en la Figura 2.6-10, el número de características (genes) de las articulaciones no es el mismo, si se muta el genotipo, se puede crear algo similar a la Figura 2.6-11; donde la articulación 2 ha pasado a tener 4 genes debido a la mutación.

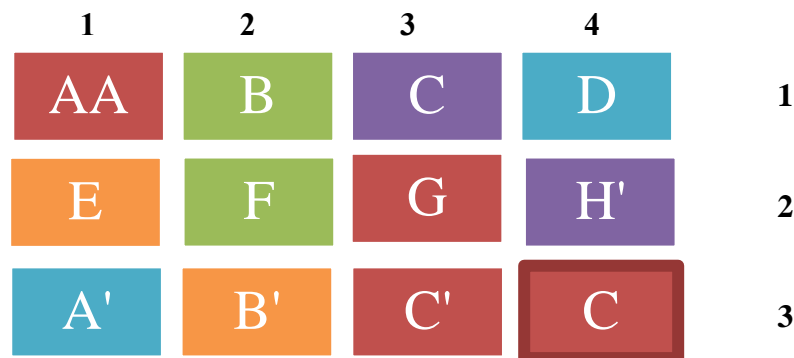


Figura 2.6-11 Ejemplo de genotipo “Articulación-Unión-Articulación” mutado

Un ejemplo práctico sería que el genotipo mutado pudiera rotar en torno a dos ejes, x e y, cuando antes sólo podía girar en torno al eje x; siendo el nuevo gen mutado el responsable de esta transformación. (Figura 2.6-12).

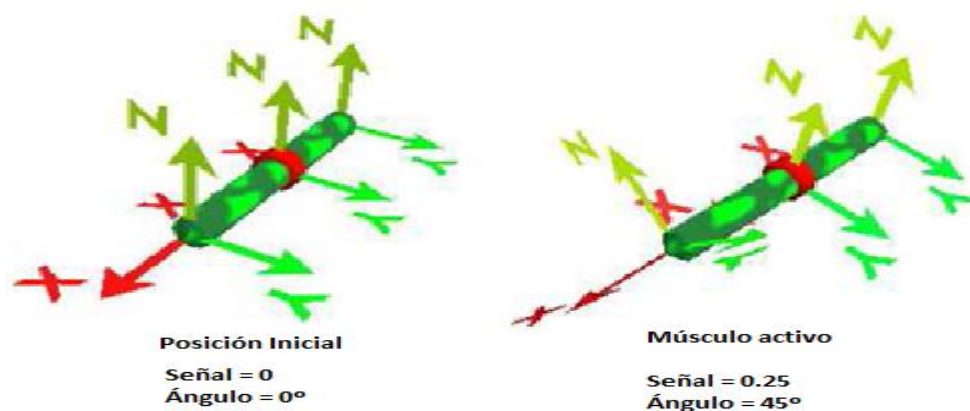


Figura 2.6-12 Ejemplo de músculo con los ejes de rotación cambiados¹⁵

Por tanto la codificación de genotipos juega un papel muy importante en su evolución, determinándola y limitándola.

En el caso particular de Framsticks, existen varios lenguajes de codificación. Estos lenguajes tienen unas características determinadas que guían el desarrollo de la evolución hacia un camino concreto con una serie de limitaciones específicas de cada uno.

Hay algunos diseñados basados en aspectos biológicos, otros que detallan más las partes mecánicas de los individuos; dependiendo del tipo de evolución objeto de experimento, se debe elegir el lenguaje de codificación adecuado para cada caso.

En este trabajo, sólo se tendrá en cuenta el lenguaje de codificación f0, por ser el que mejor detalla las propiedades morfológicas y neuronales de los genotipos y cuya sintaxis es la más sencilla y ordenada. También por ser el único compatible con FRED, el editor de genotipos de la plataforma Framsticks.

2.6.4 Formato f0

Es la sintaxis²³ básica para usuarios principiantes ya que permite de una forma rápida y sencilla la visualización, edición y creación de todos los elementos que constituyen un genotipo; también es la menos restrictiva.

Utilizando el formato f0, se notifica de manera rápida y sencilla cuando una estructura de un genotipo va a ser inviable, ya que el número de columnas, donde se indica las conexiones de cada uno de los nexos, está restringido a dos.

Gracias a la disposición de f0, se representan las variables que constituyen un genotipo mediante matrices, donde las filas corresponden a cada una de las variables y las columnas a sus propiedades, en la Tabla 2.6-2 se muestra un ejemplo. El inconveniente de este editor, es que es incompatible con el resto de codificaciones, dificultando el diseño de genotipos cuya sintaxis sea distinta de f0.

Tabla 2.6-2 Ejemplo Matriz de codificación f0

Variable	Propiedad A	Propiedad B	Propiedad C	Propiedad D	Propiedad E
Articulación A	—	3	17	56	22
Articulación B	87	—	45	89	22.3
Nexo C	64	56	—	21	69
Neurona D	37	32	91	—	18
Peso E	93	35	54	43	—

²³ Para más información sobre los distintos formatos implementados en Framsticks consultar [4] y [16]

Permite la unificación de la red neuronal y la estructura morfológica en un mismo esquema, la alteración de las propiedades y conexiones de los distintos elementos del genotipo. Se puede modificar directamente la posición de cada una de las articulaciones y nexos respecto a los tres ejes de coordenadas x, y, z. También permite la variación de las longitudes de nexos y su orientación respecto de las articulaciones a las que está unida.

2.6.5 Sintaxis

La sintaxis del formato f0 se basa en la tipificación de cada una de las variables y sus propiedades a través de unos caracteres específicos para cada variable y propiedad. Las variables que forman un genotipo están divididas en función de sus dos módulos: cuerpo y cerebro, siguiendo el formato que aparece en la siguiente Tabla 2.6-3.

Tabla 2.6-3 Sintaxis f0

Sintaxis	Variable	Módulo
P	Articulación	Cuerpo
J	Nexo	Cuerpo
C	Conexión	Cerebro
N	Neuronas	Cerebro

La forma de representación de cada una de las variables en cada fila, sigue el siguiente formato:

TipodeVariable: PropiedadA=Valor PropiedadA, PropiedadA=Valor PropiedadA,

El tipo de variable que hace referencia a cada una de las propiedades depende de las características de los distintos módulos; por ejemplo en el caso de las conexiones entre neuronas, el identificador de cada neurona, es un número entero y sin embargo la masa de las articulaciones se representa mediante números en coma flotante

Para cada una de las variables, se tienen los siguientes ejemplos de codificación:

- Articulación:

Articulación: "Posición x", "Posición y", "Posición z", "Masa", "Fricción", "Ingestión", "Asimilación",

Las propiedades que se mantienen por defecto se omiten en la codificación, por eso la longitud de las cadenas de caracteres es variable.

Ejemplo: *p:-1.19317110370129, 2.06887540496437, -0.904607765177617, ing=0.635714, as=0.121429*

- Nexos:

Nexo: "Articulación 1", "Articulación 2", "Grado de rotación alrededor del eje x", "Grado de rotación alrededor del eje y", "Grado de rotación alrededor del eje z", "Longitud", "Resistencia"

Ejemplo: $j:0, 1, rx=1.5706, rz=-2.6175, dx=1.0, 0.0, 0.0, stam=0.0892857$

- Neurona:

Neurona: "*Tipo*", "*Propiedades*"

La sintaxis de los parámetros de tipo neurona, es variable; depende de la clase de neurona y de sus propiedades. A continuación se muestran dos tipos de neuronas, la neurona sigmoidea y la neurona o músculo doblador. Como se puede observar, hacen referencia al mismo tipo de variable, pero su sintaxis sólo coincide en los primeros dígitos

Ejemplo:

Neurona o músculo doblador $n:j=0, d="/:p=1.0, r=0.0909090909090909"$

Neurona sigmoidea $n: p=0, d="N"$

- Conexiones entre neuronas:

Conexión: "*Identificador Neurona 1*", "*Identificador Neurona 2*", "*Peso de la conexión*"

Ejemplo: $c:1, 0, 0.5$

2.6.6 Genética

Conjunto de variables que afectan a la evolución del individuo, limitando y priorizando el desarrollo de determinadas características o módulos que forman cada modelo. Es fundamental para llevar a cabo la evolución de un experimento; permite seleccionar aquellas partes que se quieren desarrollar del individuo, manteniendo el resto constantes.

Es el conjunto de variables que se pueden alterar de cada individuo, está ligado al lenguaje de codificación en el que se encuentre expresado; en función del formato en el que se expresa cada individuo, se podrán alterar determinados parámetros. En este trabajo, se va a utilizar el formato f0, por ser el más sencillo e incluir en su genética los parámetros que se quieren controlar durante la evolución.

En esta plataforma se detalla el conjunto de parámetros que forman la genética como un conjunto de variables multiplicadas por determinados pesos; cuya suma proporciona el conjunto de parámetros que pueden ser utilizados por los operadores genéticos para la creación de descendencia.

El peso de cada una de ellas hace referencia a la importancia de dicha variable a la hora de ejecutarse un operador genético. Si la creación de nuevos nexos como mecanismo de evolución tiene un peso del 55 % con respecto al resto de variables; cuando se produzca la mutación de un genotipo, se puede dar el caso de que el nuevo genotipo tenga un mayor número de nexos que en el caso de que en la genética, este parámetro no fuera el dominante (tuviera un peso inferior al 55%). La genética se clasifica en tres módulos, a continuación se hace un breve resumen:

- **Módulo Físico:** Configura todo lo relacionado con las variables físicas del individuo así como su propiedades, permitiendo conservar determinadas características durante su evolución.

- **Módulo Neuronal:** Posee las mismas funciones que el módulo físico pero aplicados a los elementos que conforman la estructura neuronal del genotipo.
- **Módulo Energético:** Se encarga de configurar la cantidad de energía que poseen los genotipos en cada simulación y el aporte energético de la comida. También de establecer el tiempo de envejecimiento de los genotipos durante la simulación.

2.7 Algoritmos genéticos

2.7.1 Definición

Un algoritmo genético (AG) es la herramienta matemática que utiliza la robótica evolutiva para el desarrollo de métodos de optimización y búsqueda de diseños que proporcionen la solución adecuada de un conflicto determinado. Son un conjunto de instrucciones que indican el modo de proceder de un sistema hacia un objetivo²⁴.

Los AGs actúan sobre una población finita de individuos provocando cambios en ella; estas variaciones generan múltiples resultados, el objetivo de estos cambios es converger hacia la solución más óptima del problema.

Los individuos son el fruto de los cambios que provocan los AGs; son el conjunto de sujetos que constituyen una posible solución al problema planteado. Cada uno de ellos es un punto de exploración de una región del espacio de todas las posibles soluciones a un problema determinado.

En la Figura 2.7-1, se muestra un esquema del papel que juega cada individuo, donde cada cuadrado corresponde a una porción de espacio que podría ser la solución al problema y el conjunto de todas esas soluciones es la población de individuos que es alterada por medio de los algoritmos genéticos.



Figura 2.7-1 Ejemplo de región del espacio

A través de los AGs no se obtienen soluciones únicas, sino que proporcionan un conjunto de posibles resultados. La validez de estos resultados es evaluada mediante determinados parámetros²⁵. Estos parámetros se encargan de analizar cada solución (individuo) generado por los AGs.

Pueden ser de varios tipos, pero en el caso de este trabajo, están relacionados con el

²⁴ Toda la información en relación con los algoritmos genéticos y su aplicación en Framsticks ha sido consultada en [9], [10], [14], [15] y [17].

²⁵ Estos parámetros son definidos previamente por el creador de los AGs, personalizándolos según los conflictos que se quieran resolver.

comportamiento de los genotipos en un entorno artificial; pueden medir la velocidad de desplazamiento, la distancia recorrida etc. Por ejemplo si se quiere diseñar un modelo que transporte objetos de una habitación a otra, un parámetro sería el número de objetos que mueve cada robot y el criterio de validez del modelo sería el número mínimo de objetos que ha transportado cada prototipo.

El conjunto de resultados (individuos) generados, se valida a través de una función de evaluación; que determina el grado de aproximación a la solución del problema concreto. Dicha función depende de los parámetros citados anteriormente. Por eso cada AG está específicamente diseñado para una determinada situación, porque para cada conflicto los parámetros que se evalúan y los criterios que gobiernan los algoritmos son distintos y personalizados.

La función de evaluación genera un valor de Ajuste, único para cada individuo. Este valor se obtiene midiendo y analizando los parámetros que determinan el comportamiento del modelo. En función del resultado de las medidas, asigna un valor numérico (valor de Ajuste) que indica la aproximación del modelo a la solución del problema.

Para el desarrollo de una evolución por medio de algoritmos genéticos, en primer lugar hay que delimitar el problema y acotar el objeto de estudio, definiendo los parámetros y premisas que se analizarán de una determinada población de individuos. A continuación implementar la función de evaluación. Por último, elegir la estrategia de evolución que gobierne la supervivencia y reproducción de cada uno de los individuos de la población.

2.7.2 Símil biológico

Koza, propone la siguiente definición[6]: *“Un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suelen ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.”*

Por tanto los AGs son a la robótica lo que a la naturaleza la evolución natural, la función de evaluación es entendida como la medida de la supervivencia y reproducción de la especie y los modelos u objetos matemáticos son los distintos individuos que forman una especie.

La representación de la genética en la naturaleza es a través de los cromosomas. Como afirma Koza [6], la genética para los modelos matemáticos se ajustaría a través de cadenas de caracteres donde la relación entre ellas determinaría las distintas habilidades que serían medidas mediante la función de evaluación.

Entonces, en los objetos matemáticos²⁶ de un algoritmo genético existe un símil biológico que relaciona el cuerpo y el cerebro de un individuo con una cadena de caracteres. Estos se asemejan a los cromosomas y sus funciones a mecanismos matemáticos que simulan la reproducción sexual y asexual de la naturaleza.

²⁶ En el caso de este trabajo, los objetos matemáticos de los algoritmos serían los distintos modelos robóticos sobre los que interaccionan.

Los modelos robóticos están codificados en cadenas de caracteres de acuerdo a unas normas; esto sería el homólogo a los cromosomas en los seres vivos, los cromosomas serían las cadenas de caracteres y las normas las leyes de la naturaleza. En la naturaleza, cada ser representa un fenotipo de un determinado genotipo [3].

Sin embargo en Framsticks serían equivalentes, la instanciación del genotipo corresponde a un individuo de la población (fenotipo). No se hace distinción entre genotipo y fenotipo sino que ambos hacen referencia a un mismo individuo, él que se simula y su propia definición.

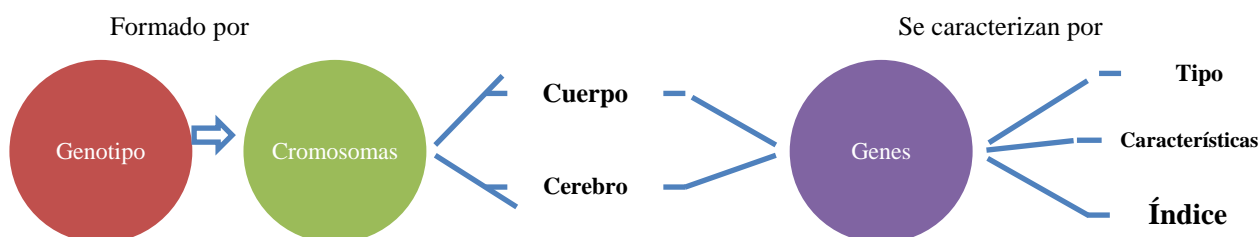


Figura 2.7-2 Esquema de las partes de un genotipo

Si todo sujeto está definido por una serie de variables que lo caracterizan, siguiendo el símil biológico, serían los cromosomas (articulaciones, nexos, neuronas y conexiones), donde los distintos genes que los forman son las propiedades de cada uno de ellos. Los genes serían los tipos (D), características(x) y posición (índice) en la cadena. En la Figura 2.7-2 y Figura 2.7-3²⁷, se muestra un esquema de las partes que componen cada sujeto y cromosoma.

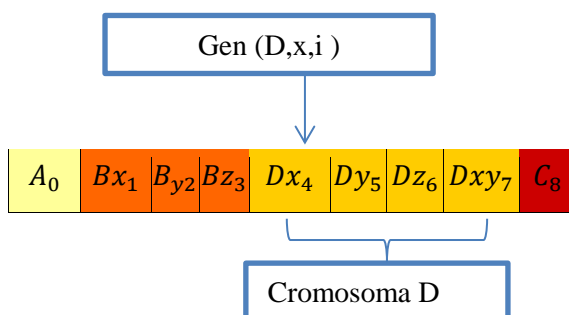


Figura 2.7-3 Estructura de un cromosoma

Un genotipo está formado por distintos cromosomas, pertenecientes al cuerpo y cerebro. Cada uno de estos tipos está formado por un conjunto de genes. Los cromosomas son los distintos elementos que forman el cerebro (neuronas, conexiones) o el cuerpo (nexos, articulaciones). Los genes guardan información sobre las características de cada elemento y están numerados, cada uno de ellos caracteriza a los distintos cromosomas que forman un sujeto. La disposición de cada uno de los genes y cromosomas influye en la estructura del individuo y se representa a través de las distintas conexiones que existen entre los cromosomas.

Por tanto la representación de los individuos sobre los que interaccionan los AGs guarda similitud con la disposición de los cromosomas y genes en la naturaleza. Las cadenas de

²⁷ Donde cada celda es un gen y cada conjunto de celdas del mismo color representa un cromosoma.

caracteres serían el equivalente a las cadenas de ADN de los seres vivos.

En el mismo contexto, en la aplicación de los AGs existe otra analogía con la naturaleza. El objetivo de los algoritmos es encontrar el conjunto de resultados que más se adecúen a la solución de un problema, el sujeto ó resultado que presenta el mejor valor de Ajuste es el que se mantiene en la población y genera la descendencia. Paralelamente, en la naturaleza, la evolución tiene como objetivo la supervivencia de la especie, entendida como la adaptación al medio. Es decir, el individuo que mejor se adapte al entorno tendrá más posibilidades de sobrevivir y continuar la especie; miden el éxito de un sujeto al competir por determinados recursos con respecto al resto de miembros.

Llegados a este punto la supervivencia guarda una relación directa con el valor de Ajuste de los algoritmos ya que ambas tienen como objetivo la adaptación del sujeto al medio, a los distintos entornos dinámicos u estáticos con los que se pueda encontrar. En los dos casos se mide el progreso de cada sujeto en su intento de acomodarse al medio y explotar los recursos del mismo como método de evaluación.

2.7.3 Algoritmos genéticos aplicados en Framsticks

Uno de los pilares fundamentales de la herramienta es el estudio de la evolución de distintos modelos, aplicando diferentes criterios de selección, alterando sus propiedades e intercambiando cromosomas entre individuos. Su función es la ejecución de mecanismos que provoquen la evolución de los prototipos hacia un objetivo fijado por el usuario.

Son mecanismos encargados de la transformación, reproducción y selección de los distintos individuos que mejor contribuyen a la realización de una tarea u objetivo concreto. Dependiendo de la función que realicen, se distinguen tres tipos: Algoritmos de Selección y reemplazo, Operadores Genéticos (OG) y Algoritmos de Evaluación²⁸.

Los primeros se encargan de determinar qué individuos formarán parte de la siguiente población o serán los progenitores de la misma. Los segundos (OG) de crear la descendencia de los individuos seleccionados; con esta intención, actúan sobre los genotipos (Progenitores) modificando sus propiedades mediante el intercambio de sus distintos cromosomas para la creación de nuevos individuos (Descendencia). Y los últimos algoritmos analizan los modelos y determinan su posición global con respecto del resto de la población (Función de evaluación).

2.7.4 Algoritmos genéticos de selección

Mediante los algoritmos de selección, se eligen los sujetos que pueden generar descendencia²⁹. Si la presión de selección de individuos es muy alta, sólo pasarán a la siguiente generación los mejores individuos y la búsqueda del conjunto de resultados más óptimo se centrará en un entorno próximo a las características de los mejores individuos actuales. Provocando una reducción de la probabilidad de exploración de determinadas regiones.

²⁸ Todos los AG pueden ser configurados por el usuario a través de la herramienta, en los sucesivos capítulos se detalla como hacerlo.

²⁹ En un contexto biológico sería equivalente a los individuos que mejor se adaptan y sobreviven son los elegidos para reproducirse.

Si se elimina la probabilidad de que los individuos menos aptos se reproduzcan, la evolución converge en una población homogénea a medida que aumentan las generaciones. Aumentando la capacidad del conjunto de individuos que se selecciona, se modifica la presión de selección, convergiendo en un aumento de la probabilidad de selección de los peores sujetos. Al haber más sujetos la probabilidad de elegir a los peores individuos aumenta, garantizando la diversidad de la población.

Existen varios criterios para seleccionar a los progenitores, en este trabajo se evalúan los siguientes³⁰:

- **Selección aleatoria:** Los modelos son seleccionados al azar, no se sigue ningún criterio.
- **Ajuste proporcional (Ruleta ó selección de Montecarlo):** se concede a cada uno de los sujetos un valor entre el $[0...1]$, en relación con su Ajuste. Dicho valor equivale a la porción que le correspondería de una ruleta, los individuos con valores más altos les corresponderá mayor extensión de la ruleta (Figura 2.7-4). Para escoger el individuo se genera un número aleatorio comprendido entre $[0...1]$, se recorre la ruleta sumando todos los valores de los individuos hasta alcanzar el valor seleccionado y el sujeto que ocupe esa posición es el seleccionado. A medida que aumenta el tamaño de la población se vuelve cada vez más ineficiente, ya que la porción que ocupa cada sujeto va disminuyendo aumentando las probabilidades de elección de los peores individuos.

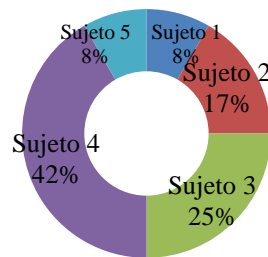


Figura 2.7-4 Ejemplo de Proporción de Ajuste (Ruleta)

- **Torneo:** Se selecciona el sujeto mediante comparación directa de su valor de Ajuste con respecto al de un conjunto de sujetos. Dependiendo del criterio utilizado en la comparación, hay dos tipos de torneos, los deterministas, en los cuales el modelo que presenta el mejor valor de Ajuste es el seleccionado, y los probabilísticos³¹.
- **Evaluación múltiple:** Es una variante del torneo, se evalúa cada individuo más de una vez, se introduce el mismo sujeto más de una vez en la simulación, el valor de Ajuste se obtiene como media de las evaluaciones y se selecciona el mejor.³²

³⁰ Toda la información sobre los distintos tipos de AGs se encuentra detallada en [9] y [10]

³¹ En este tipo de torneos, se genera un número aleatorio dentro del intervalo $[0...1]$, si el número originado es mayor que un parámetro predefinido “a”, el cuál suele tomar valores en el rango $0,5 < a < 1$. Se selecciona al sujeto que tenga el mayor valor de “a” y en caso contrario, el menor.

³² Esta variante está desarrollada en Framsticks y se denomina *evaluación múltiple* pero no será utilizada en este trabajo.

2.7.5 Algoritmos genéticos de reemplazo (Eliminación de individuos)

Se utilizan cuando existe una única población limitada, es decir cada nuevo individuo “ocupa” el lugar de otro o se ha llegado al límite de la capacidad de la población y por tanto para introducir nuevos individuos hay que eliminar algunos. Para llevar a cabo esta operación, se utilizan los Algoritmos de reemplazo. Existen diversos tipos, en Framsticks se utilizan los siguientes:

- **Ajuste inversamente proporcional:** Cuanto mayor es el Ajuste, menor es la posibilidad de ser eliminado y “reemplazado”. Los sujetos que tienen peor Ajuste comparado con el resto de las instancias, son eliminados de la población.
- **Reemplazo aleatorio:** se inserta el nuevo individuo, sustituyendo a otro aleatoriamente.
- **Reemplazo de los peores:** Los individuos con el valor de Ajuste mínimo son eliminados.

2.7.6 Operadores Genéticos (OGs)

Encargados de la creación de nueva descendencia a partir de uno o más individuos. Su función equivale a la reproducción sexual y asexual de los seres vivos. Los tres tipos de OG que se utilizan en Framsticks, se describen a continuación:

2.7.6.1 Cruce

Operador Genético cuyo principio está basado en el intercambio de atributos (genes) entre dos individuos de una misma población. Cada uno de los miembros de la pareja aporta sus atributos y estos se recombinan con su pareja para crear un nuevo sujeto.

En la siguiente figura (Figura 2.7-5) se muestra un ejemplo de como se lleva a cabo este proceso, para observar un ejemplo de algoritmo de cruce siguiendo la codificación de Framsticks, en la Figura 2.6-9 se muestra un ejemplo.

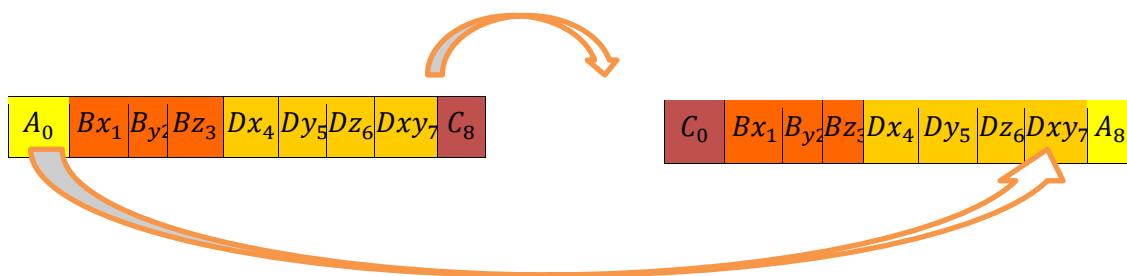


Figura 2.7-5 Cruce de cromosomas

Si uno de los padres tiene un Ajuste alto y sus cromosomas se cruzan con otro progenitor dando lugar a un nuevo individuo, los hijos pueden heredar los genes que provocan los altos valores de Ajuste. En el caso de dos padres con valores altos de Ajuste, significa que sus genes poseen determinadas características que incrementan el Ajuste, por lo que los hijos tienen más probabilidades de heredar unos cromosomas más aptos que los hijos de padres cuyos valores de Ajuste son peores.

Por el contrario, si los hijos heredan los genes que no contribuyen al aumento del Ajuste, la descendencia será peor que los padres; en estos casos se debe aplicar un criterio de selección

que evalúe tanto a hijos como padres para así elegir a los mejores individuos que pasarían a la siguiente generación. En el caso de que los hijos “malos” pasaran a la siguiente generación, existiría la posibilidad de mantener los genes “buenos” de los padres y en los sucesivos cruces produjeran individuos con mejores valores de Ajuste. Por eso hay que favorecer la diversidad y evitar la homogeneidad de la población.

Hay diversos tipos de OG de cruce dependiendo del método utilizado para intercambiar los genes, cruce de 1 punto, cruce de 2 puntos, cruce uniforme. Son los más utilizados, en la Figura 2.7-6, Figura 2.7-7 y Figura 2.7-8 se explica de manera visual cada uno de los tipos mencionados:

- **Cruce de 1 punto:**

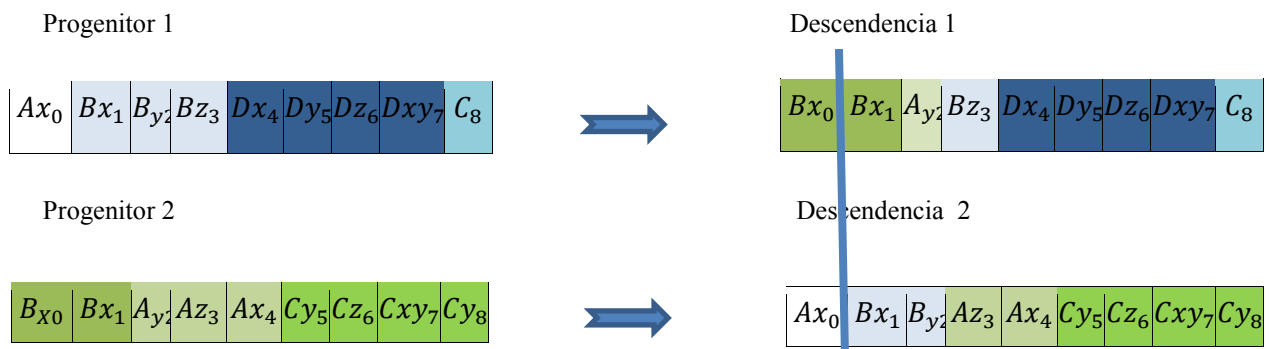


Figura 2.7-6 Cruce de un punto

- **Cruce de dos puntos:**

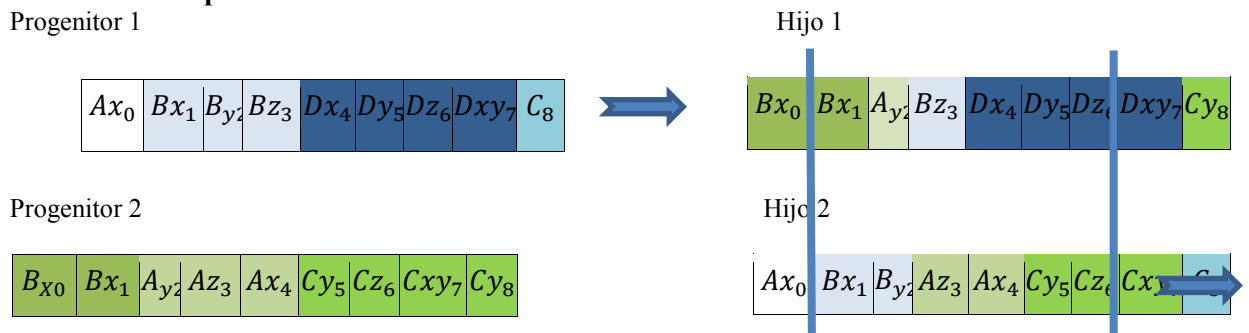


Figura 2.7-7 Cruce de dos puntos

- **Cruce Uniforme:** Consiste en la creación de una máscara que determina el origen de cada uno de los genes, mediante esta técnica la probabilidad de pertenecer a uno u otro progenitor es la misma. Uno de los métodos más utilizados es el desarrollo de una máscara codificada en valores binarios y dependiendo del valor de cada una de las posiciones, el hijo heredará el gen del padre 1 o del padre 2, a continuación(Figura 2.7-8) se muestra un ejemplo de este tipo de cruce:

Máscara de cruce:

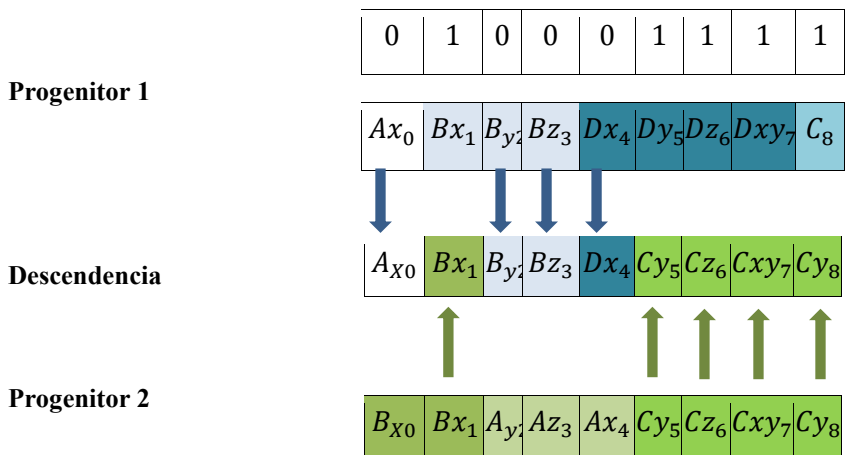


Figura 2.7-8 Ejemplo de cruce uniforme³³

- **Cruces específicos de combinaciones no binarias:** En el caso de utilizar genotipos compuestos por valores numéricos o que posean distintos grados de aportación(pesos) en función de distintas características, se pueden utilizar otras técnicas, como por ejemplo:
 - **Media:** La descendencia tiene como resultado, la media de los valores de los cromosomas de los padres, por lo que sólo es posible generar un descendiente. En la siguiente tabla (Tabla 2.7-1) se muestra un ejemplo numérico:

Tabla 2.7-1 Ejemplo de cruce de media

Cromosoma	Progenitor 1	Progenitor 2	Descendiente
A	0.5	0.2	0,35
B	0.2	0.1	0,15
C	0.1	0.1	0,1
D	0.3	0.6	0,45

- **Media geométrica:** Cada cromosoma de la descendencia se obtiene del valor de la raíz cuadrada del producto de cada uno de los cromosomas de los progenitores, el valor del cromosoma A del hijo se halla aplicando la siguiente Ecuación 2.7-1, en la Tabla 2.7-2 se encuentra un ejemplo.

$$A_H = \sqrt{A_{P1} * A_{P2}}$$

Ecuación 2.7-1 Media Geométrica³⁴

³³ El bit 1 representa a los cromosomas del Progenitor 2, mientras que el bit 0, los del Progenitor 1.

³⁴ Donde A_H pertenece al hijo y A_{P1} , A_{P2} a cada uno de los padres.

Tabla 2.7-2 Ejemplo de cruce geométrico

Cromosoma	Progenitor 1	Progenitor 2	Descendiente
A	0.5	0.2	0,31622777
B	0.2	0.1	0,14142136
C	0.1	0.1	0,1
D	0.3	0.6	0,42426407

2.7.6.2 Mutación

Estrategia reproductiva que modifica uno o varios cromosomas de un sujeto aleatoriamente. Altera las propiedades de un individuo provocando variaciones en su Ajuste, estas modificaciones pueden incrementar su valor o disminuirlo dependiendo de los cromosomas que altere y los objetivos de la evolución.

Genera a partir de un individuo “padre”, su descendencia con una estructura similar a la del progenitor. A través de este proceso se garantiza que ninguna de las variables que forman parte de los individuos tenga una probabilidad nula de ser modificada. Por ejemplo, si los parámetros que conforman la estructura de un sujeto son los representados en la Figura 2.7-9, donde cada una de las partes del individuo representa un cuadrado y las variables contenidas en ese cuadrado son las propiedades³⁵.

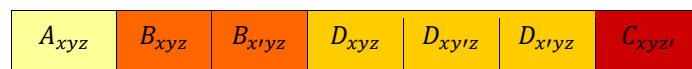


Figura 2.7-9 Ejemplo de estructura de un cromosoma

Si sólo aplicamos el operador genético de cruce, existe la probabilidad de que una de ellas no modifique su posición y por lo tanto no forme parte del proceso evolutivo; sin embargo al mutar el sujeto, aseguras que todas los parámetros del individuo intervengan en la evolución.

Aunque la mutación consiste en un reemplazo aleatorio de variables, es posible modificar los valores de un gen multiplicándolo por un valor contenido entre el intervalo $[0...1]$. En la Tabla 2.7-3 se observa un ejemplo de una mutación de cromosomas aleatoria:

Tabla 2.7-3 Mutación

Cromosoma	Progenitor	Descendencia
A	0.5	0,3
B	0.2	0,12
C	0.1	0,06
D	0.3	0,18

³⁵ En los apartados anteriores se detalla la estructura de los genotipos en relación con los AG.

Si se quieren controlar y limitar las variables que pueden ser modificadas por medio de la mutación hay que recurrir a la genética³⁶. Mediante la genética se puede limitar la mutación de un individuo a un solo parámetro, en la Figura 2.6-10 y Figura 2.6-11 se encuentra un ejemplo de esta limitación. En la siguiente tabla, Tabla 2.7-4, se muestra un ejemplo de mutación por multiplicación de una constante, la fórmula utilizada se describe a continuación, Ecuación 2.7-2:

$$A_H = A_{P1} * C$$

Ecuación 2.7-2 Mutación de genotipos

Tabla 2.7-4 Ejemplo de mutación con permutación del gen A

Cromosoma	Progenitor	Descendencia
A	0.5	0,5
B	0.2	0,12
C	0.1	0,06
D	0.3	0,18

2.7.6.3 Copia

El operador genético copia es otro método utilizado en los AG para la creación de un nuevo individuo a partir de otro anterior. Se obtiene un nuevo sujeto cuyos cromosomas son idénticos a los de su progenitor.

El peso de este operador en el AG, deber ser el menor en relación con el resto de operadores, ya que si se realizan muchas copias de individuos la población tiende a converger hacia un individuo concreto y el Ajuste converge hacia valores similares a los del individuo copiado. Lo que implica que la búsqueda en el espacio del problema se focaliza en un individuo concreto y no se produce la transformación genética en nuevos modelos. A continuación, en la Figura 2.7-10, se representa un ejemplo:

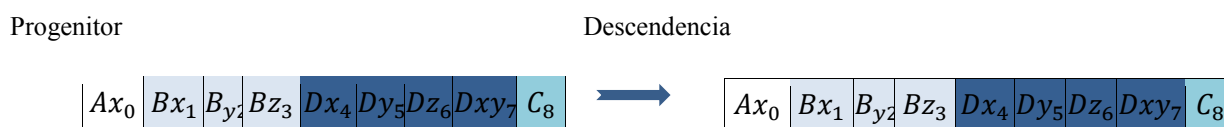


Figura 2.7-10 Ejemplo de algoritmo de Cruce

2.7.7 Algoritmos genéticos de evaluación

Los algoritmos genéticos de evaluación se encargan de validar cada una de las instancias como posible solución al problema. Determinan si el comportamiento del individuo es favorable a la evolución o si por el contrario, las capacidades o comportamientos que ha ejecutado no lo son.

³⁶ La genética se encarga de controlar qué variables y partes del sujeto pueden ser alteradas durante su evolución. Para más información consultar 2.6.6.

A través de esta función se obtiene un valor numérico denominado Ajuste, único para cada miembro de la población que es evaluado. El Ajuste es la aproximación de cada instancia a la solución del problema, se obtiene de la suma ponderada de las variables³⁷ que se evalúan. Para calcularlo se utiliza la siguiente fórmula (Ecuación 2.7-3):

$$Ajuste = \sum_{w=0}^{w=1} w * j$$

Ecuación 2.7-3 Función de evaluación

Donde w es el valor de cada uno de los pesos y cuya suma es igual a 1, y j representa a cada una de las variables que comprenden el Ajuste³⁸.

En Framsticks, el valor de Ajuste equivale al resultado de la evaluación del comportamiento del genotipo simulado. Por ejemplo, si la función de evaluación depende sólo de la variable velocidad. El robot cuya velocidad haya sido mayor, tendrá un valor de Ajuste mayor y por tanto, si el objetivo de la evolución era desarrollar el individuo más rápido, todas las cualidades de dicho genotipo serán las más apropiadas para las siguientes generaciones.

Existen dos valores de Ajuste, el valor real del Ajuste y el valor final del Ajuste, conocido como Ajuste Final. El valor real es el resultado de la función de evaluación con los parámetros elegidos por el creador de la simulación y no tiene en cuenta el resto de individuos de la población. Sin embargo Ajuste Final es el Ajuste de un genotipo individual con respecto al resto de genotipos que han sido evaluados durante la simulación; mediante este valor se obtiene la posición global del genotipo en la población, el ranking de los mejores individuos. Para obtenerlo, se aplica en la Ecuación 2.7-4:

$$Ajuste Final = \left[\left(\frac{\sum_{i=0}^{i=n} Ajuste_i}{n} \right) - n \right] * \sigma$$

Ecuación 2.7-4 Ajuste final

Donde n es el número de individuos que hay en el conjunto de genotipos, i cada uno de ellos y σ la desviación estándar.

2.7.8 Limitaciones de los algoritmos genéticos

Los algoritmos genéticos se encargan de la búsqueda de la solución más óptima, muestran cada modelo como solución a un problema dependiendo de su función de evaluación, componen el análisis del comportamiento de un individuo en relación con una determinada habilidad y comparándolo con el estudio del comportamiento del resto de individuos que participan en el experimento.

Las principales limitaciones son debidas al método de elección del mejor individuo. El más utilizado es la comparación del comportamiento y habilidades de un individuo con otro,

³⁷ El conjunto de variables que pueden participar en la función de evaluación se detallan en [4]

³⁸ Si la suma de todos los pesos no es igual a 1, existe una función llamada *criteria normalization*, disponible tanto en los scripts como en la GUI, que ajusta automáticamente dichos pesos para que la suma de todos ellos sea 1.

provocando la elección del que mejor cumpla los objetivos. Sin embargo esta elección puede no ser la más óptima; si todas las posibles soluciones evaluadas no han cubierto todas las regiones del espacio disponibles (Figura 2.7-11), la solución idónea no ha sido contemplada y por tanto el resultado obtenido no es el mejor.

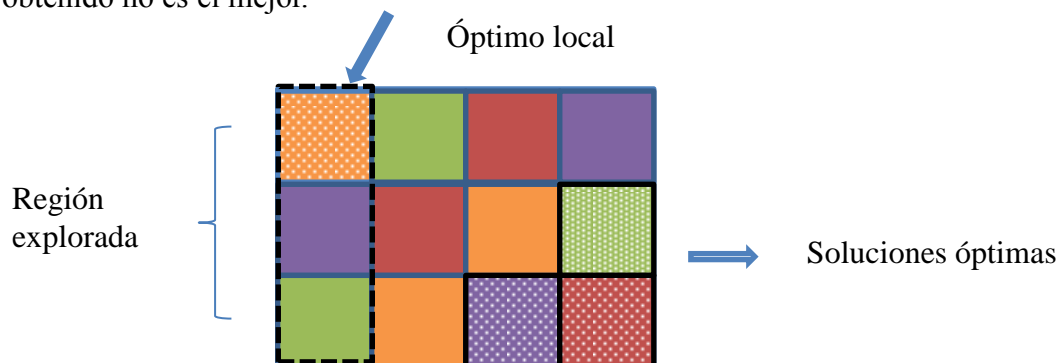


Figura 2.7-11 Esquema de la disposición de las soluciones y las regiones exploradas

El “mejor” individuo se convierte en un óptimo local, guiando la evolución hacia un conjunto de individuos que han implementado ciertas habilidades que, comparadas con el resto, aumentan la función de evaluación. Pero de cara al conjunto global, no constituyen la solución global más óptima; por tanto el óptimo global no se alcanzaría ya que no se reproducirían todas las posibles soluciones y se daría por buenas soluciones parciales.

El estancamiento está también relacionado con lo anterior; si se produce un óptimo local la evolución se estancaría al no poder explorar nuevas regiones del espacio debido a que los “mejores” individuos de la población están concentrados en una determinada porción y no se exploran otras zonas que podrían tener mejores sujetos que incrementaran el valor de Ajuste. La evolución pasado un tiempo se detendrá por no haber individuos que aporten nuevas habilidades y propiedades que aumenten la diversidad en la población e incrementen el valor de Ajuste.

La limitación de la evolución por la codificación de las propiedades de los individuos es otro de los problemas, en el apartado 2.6.6 se explica con detalle esta situación.

Mediante el empleo de los AGs se desarrollan métodos de búsqueda de soluciones a un conflicto que se plantea, cuyo resultado, en ocasiones, puede ser imposible de realizar, debido a las limitaciones del mundo físico. En muchos casos se obtienen soluciones a determinados problemas que no pueden llevarse a cabo en la realidad por diversas limitaciones. Es el caso del experimento de Ingo Rechenberg, mediante el empleo de algoritmos evolutivos optimizó el diseño de las tuberías para que el flujo de entrada fuera mayor, Figura 2.7-12 (a), aunque luego por causas económicas³⁹ no se pusiera en práctica. Sin embargo la antena⁴⁰ usada en el satélite NASA ST-5 creada Jason Lohn y sus compañeros de la NASA que también fue diseñada empleando algoritmos genéticos, Figura 2.7-12 (b), sí que fue utilizada.

³⁹ El coste de producción de este modelo era superior al convencional por lo que se rechazó su comercialización. Para más información consultar [18]

⁴⁰ Para más información consultar [8]

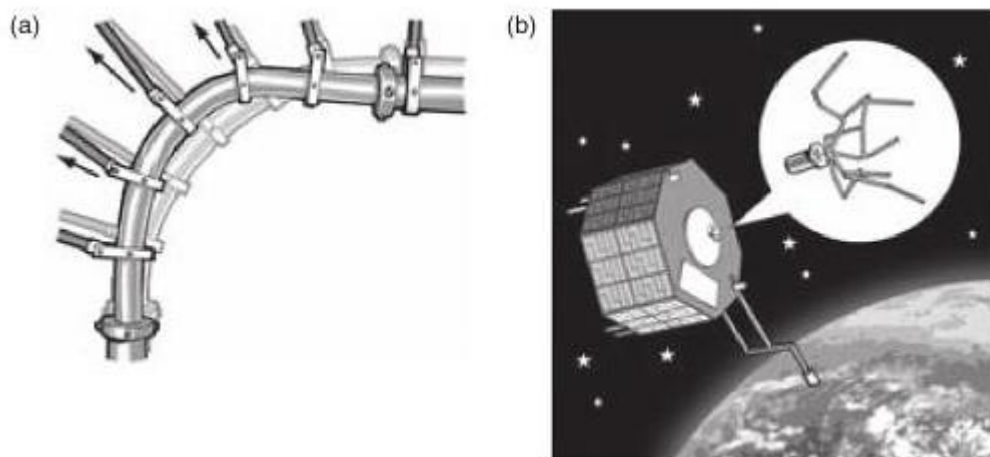


Figura 2.7-12 Ejemplo ilustrado⁴¹

2.8 Simulación artificial evolutiva

Las simulaciones artificiales evolutivas es otro de los pilares fundamentales de la plataforma, permiten observar el comportamiento de los genotipos y como interactúan en distintos entornos artificiales con el resto de genotipos durante la ejecución del proceso evolutivo.

Consisten en la representación del comportamiento de determinados individuos en un mundo virtual. El estudio de su comportamiento será analizado y dependiendo del objetivo y características de la simulación, dichos individuos evolucionarán a nuevos modelos que mejoren las características y habilidades de sus antecesores.

En una simulación se lleva a cabo un proceso evolutivo, a través de la creación, análisis y evaluación del comportamiento de los objetos simulados. El objetivo de una simulación depende del proceso evolutivo que se ejecute en ella, es decir depende de la función de evaluación que rija la evolución. Dependiendo de las características de esta función tendremos dos tipos de simulaciones que se explican a continuación.

En Framsticks, las simulaciones estan formadas por un conjunto de genotipos divididos en dos grupos: conjunto de genotipos y población temporal; donde se llevan a cabo las distintas transformaciones de los genotipos. Las interacciones entre ambos grupos y de cada uno de forma individual están gobernadas por los AGs, encargados también del avance y control de la simulación.

A continuación, en la Figura 2.8-1, se muestra un esquema del flujo de la simulación, identificando los dos módulos que componen la simulación evolutiva:

⁴¹ Imágenes obtenidas de [7]

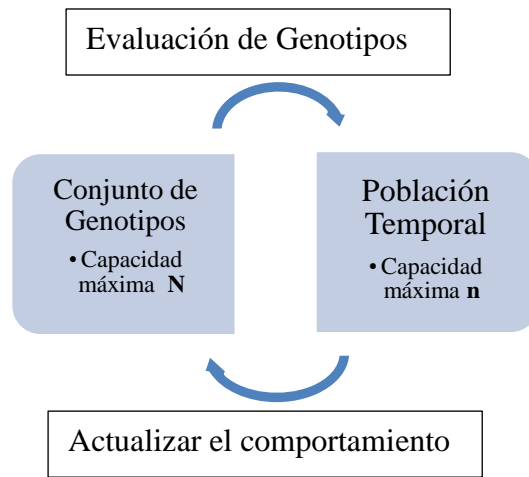


Figura 2.8-1 Ejemplo de Flujo de una simulación evolutiva

Los parámetros clave que el paso de un módulo a otro durante la simulación evolutiva relacionados con la capacidad de los conjuntos de genotipos y la función de evaluación.

A menor capacidad de individuos (N) la presión de selección es mayor y son escogidos aquellos individuos cuyos resultados en la simulación cumplen una serie de condiciones relacionadas con la función de evaluación. Sin embargo, si el valor de $N > N'$ ⁴², la presión será menor y por tanto, todos los individuos simulados pasarán a formar parte del conjunto de genotipos y no se ejecutará ninguna regla de selección hasta que la población N se haya completado.

Los mecanismos encargados de regular la selección y eliminación de genotipos en el conjunto, son los AGs de selección y reemplazo. Para añadir individuos a la población temporal ($N > N'$), los miembros del conjunto de genotipos se seleccionan utilizando algoritmos de selección y son transformados en función de los operadores genéticos, tales como, la mutación, el reemplazo etc.

Una vez que estos nuevos individuos han sido simulados y se ha calculado el valor que aportan a la función de evaluación, son introducidos en el conjunto de genotipos. Si $N = N'$, la capacidad máxima del conjunto de genotipos se ha alcanzado y se han de eliminar genotipos; se ponen en ejecución los algoritmos de eliminación de genotipos para reducir la población y que permanezcan los individuos cuya función de evaluación se Ajuste más a lo especificado en los AGs.

2.8.1 Tipos de simulaciones

Como el objetivo de una simulación evolutiva depende de su función de evaluación, hay dos tipos distintos de simulaciones, las simulaciones espontáneas y simulaciones directas.

Las simulaciones espontáneas tienen como función de evaluación la supervivencia del genotipo, no están sujetas a ningún criterio que intervenga sobre las habilidades de los individuos. Los experimentos son simulados y se observa su comportamiento e interacción con el entorno. Es un símil de la evolución natural porque la elección de un modelo frente a otro se basa en su supervivencia en el entorno y no en la optimización de una habilidad concreta.

⁴² N' hace referencia al número actual de individuos del Conjunto de Genotipos, cuyo valor no puede ser mayor que el de la Capacidad(N).

Sin embargo en las simulaciones directas, objeto de estudio de este trabajo, el usuario predefine la meta de la simulación. Los genotipos creados evolucionan siguiendo una línea determinada que provoca la mejora de determinadas capacidades que se acerquen a la meta de la simulación. Para poder configurar dichos objetivos está la función de evaluación, encargada de asignar un valor determinado en función de los resultados de los parámetros, obtenidos en la simulación. En la Figura 2.8-2, se muestra un esquema del proceso de una simulación directa:

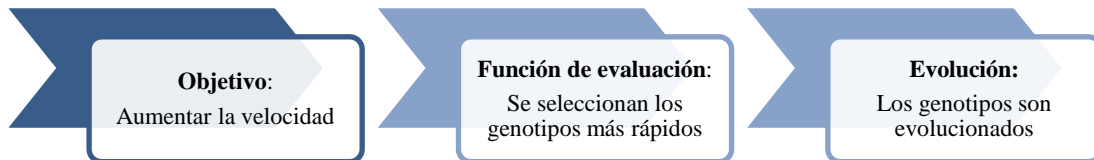


Figura 2.8-2 Simulación evolutiva directa

En este caso (Figura 2.8-2), el objetivo de la simulación evolutiva es aumentar la velocidad de los robots; con esta intención se configura la función de evaluación para que dependa de los parámetros que miden el incremento de velocidad (distancia recorrida, duración del recorrido etc). Por medio de esta función se escogen los modelos más rápidos; a continuación, estos modelos son evolucionados para que a la siguiente generación sean transferidas las propiedades que han hecho más veloces a sus padres.

Por el contrario, si se tratase de una simulación indirecta, el criterio de selección de los progenitores estaría basado en la supervivencia y adaptación al medio y por tanto las nuevas generaciones de genotipos podrían ser más lentas que sus progenitores.

Entonces para satisfacer unos objetivos evolutivos determinados y distintos de la supervivencia de los genotipos, se deben utilizar las simulaciones evolutivas directas.

2.8.2 Estructura de la evolución

En este TFG, se entiende como evolución al proceso mediante el cual los robots de un experimento son sometidos a transformaciones para mejorar o configurar determinadas habilidades; la evolución provoca la selección, eliminación y generación de descendencia a través del empleo de diferentes AGs. El tipo, prioridad y orden de los AGs esta determinado por las distintas estrategias evolutivas que se pueden dar en la evolución.

Existen diversas estrategias de evolución, que determinan la selección, eliminación y creación de los individuos en función de distintos algoritmos genéticos y la capacidad de la población. Estas estrategias son las premisas que se ha de fijar para cada uno de los AGs que participan en la evolución. Uno de los pioneros en esta investigación fue el profesor alemán Ingo Rechenberg, cuyas publicaciones se han utilizado por números investigadores, como base para la investigación sobre los AGs y su impacto en la evolución.

Las estrategias evolutivas hacen referencia a las distintas políticas seguidas a la hora de elegir los AGs que participan en la evolución y las premisas y criterios que los caracterizan. Están ligadas a la prioridad y tipo de algoritmo que participa en la evolución, generando procesos en los que la mutación adquiere importancia, otros más relacionados con la selección de individuos etc. En este trabajo se emplean varias estrategias evolutivas que influyen en la evolución de los genotipos; se realizan varios experimentos modificando la presión de selección de los genotipos y variando los procesos de generación de descendencia (OGs).

Los algoritmos genéticos son el conjunto de instrucciones que guían el desarrollo de una evolución hacia un objetivo concreto, la ejecución de estas órdenes sigue unos determinados pasos. En este apartado se detallan los pasos que se llevan a cabo durante todo el proceso evolutivo.

Antes de realizar el proceso evolutivo, hay que elegir los AGs que van a participar en la evolución, en las siguientes figuras (Figura 2.8-3 y Figura 2.8-4) se muestra un ejemplo de una evolución y de los distintos AGs que participan en ella, indicando el orden y criterio de realización.

Se deben especificar también los criterios de finalización de cada una de las premisas de los algoritmos. Son de vital importancia ya que ejercen un papel importante sobre la evolución. Estos criterios influyen en la función evaluación y la presión de selección; que a su vez determinan la capacidad de la población, el criterio de terminación y la elección de los mejores individuos de la población.

Para realizar un proceso evolutivo, el primer criterio que hay que fijar, es la función de evaluación⁴³, dependiendo de los valores obtenidos por cada individuo, la evolución seguirá su curso o se interrumpirá en el caso de que se cree un individuo o conjunto de individuo que cumpla con las premisas definidas.

A continuación, se debe fijar la capacidad de la población que va a ser evaluada y la que va a ser simulada; que luego influirá en los criterios de selección y reemplazo de individuos y que está directamente relacionada con la presión de selección. Si la población ha llegado a su límite de capacidad y se han creado nuevos individuos, se tendrá que hacer una selección de aquellos individuos que pueden continuar en la población y los que deben ser eliminados; terminado este proceso, se creará la nueva generación de individuos. El siguiente paso es configurar los operadores genéticos que van a ejecutarse sobre la población y sus prioridades para la creación de la descendencia.

Conocidas las premisas y principales características, en la Figura 2.8-3, se muestra un ejemplo de la estructura y en la Figura 2.8-4, se encuentra un ejemplo de la estructura escrita en pseudocódigo de un conjunto de AGs.

⁴³ El objetivo de la evolución y de la ejecución de los AGs está contenido en la función de evaluación mediante los distintos parámetros que la forman y la ponderación de cada uno de ellos.

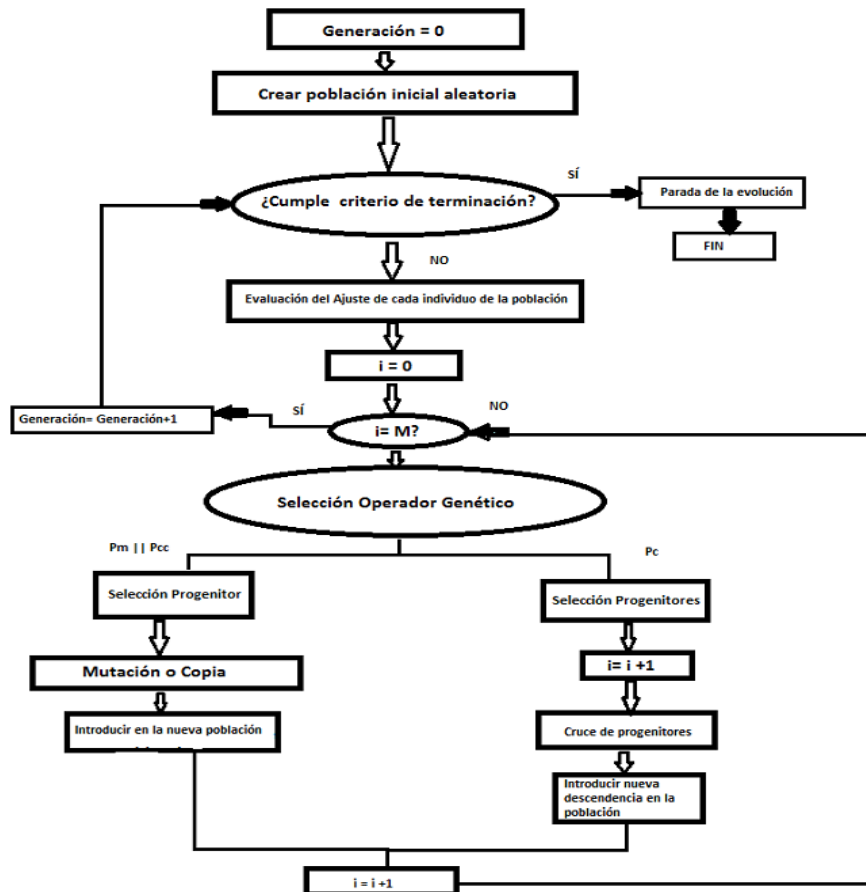


Figura 2.8-3 Ejemplo traducido del libro de J, Koza [6]

En la Figura 2.8-3, se encuentra representada un ejemplo de evolución basada en el empleo de los AGs, en esta evolución hay una población de individuos inicial que es evaluada, si cumple los objetivos se termina la evolución, en caso contrario, se genera nueva descendencia a partir de los operadores genéticos y son introducidos en la nueva población. Si la capacidad de la población no ha sido superada, se siguen creando nuevos genotipos.

Cuando se supera el límite de capacidad, se pasa a la siguiente generación de genotipos y los individuos son evaluados; si cumplen los objetivos se termina la simulación, si no son cumplidos, se seleccionan a los progenitores de la siguiente generación. Y así hasta que el criterio de satisfacción se cumpla.

Por otro lado, en la Figura 2.8-4, se encuentra un ejemplo de evolución escrito en pseudocódigo; la evolución sigue el mismo comportamiento que en el caso anterior con la excepción del empleo de un nuevo algoritmo, el algoritmo de reemplazo. Para introducir un nuevo sujeto a la población se evalúan todos los individuos y si son mejores que los padres, los descendientes son introducidos en la población, en caso contrario, los padres permanecen.

```
Iniciar población actual
```

```
while1( Ajuste no cumpla los objetivos de AG)
  iniciar nueva población temp
  while2(temp< capacidad máxima temp)
    seleccionar padres
    descendientes=operadores genéticos*padres
    evaluar descendientes
    if(Ajuste descendientes mejor que Ajuste padres)
      añadir descendientes a población temp
    else
      añadir padres a población temp
  End while2
  Generación población actual +1
  población actual = población temp
End while1
```

Figura 2.8-4 Ejemplo pseudocódigo

Una vez configurados todos los criterios que afectan a los AGs y mostrado algunos ejemplos, se detalla el proceso evolutivo que se lleva a cabo en este TFG:

Se crea una población de individuos, se simulan y se analiza su comportamiento atendiendo a los criterios de la función de evaluación. Si cumplen los objetivos establecidos, la simulación se detiene ya que se han obtenido las soluciones al problema. Si no se cumplen, hay que generar nueva descendencia, por tanto se eligen los progenitores de la nueva población, siguiendo los criterios de los algoritmos de selección.

Sobre los progenitores operan los OG, generando nuevos individuos a partir de uno o más progenitores. Si estos individuos son válidos, se simulan y se estudia su comportamiento. En el caso de no haber llegado al máximo de la capacidad de la población, los nuevos genotipos son introducidos en la población. De lo contrario, se hace una selección de los individuos que van a permanecer en la población, dando lugar a la siguiente generación. Para ello se evalúan todos los individuos y se aplican los algoritmos de reemplazo, para determinar qué genotipos permanecen en la población y cuales son eliminados.

Si durante la evaluación de los genotipos se cumplieran los criterios de la función de evaluación, se detendría la simulación. En caso contrario se volvería a realizar la selección de los mejores individuos, que generarían la nueva descendencia, serían evaluados todos de nuevo y así hasta conseguir los resultados requeridos, es decir el valor de Ajuste deseado.

CAPÍTULO 3

Framsticks GUI

3.1 *Definición y conceptos*

La Interfaz gráfica de usuario (GUI) de Framsticks es la pieza fundamental de la plataforma, engloba todas las funciones que pueden ejecutarse en la herramienta⁴⁴. Dirigida tanto a usuarios avanzados como principiantes, sus simulaciones y genotipos predefinidos ayudan a los usuarios principiantes a tomar contacto con la herramienta; para usuarios más avanzados en el manejo de la aplicación, ofrece la posibilidad de crear y configurar tus propias simulaciones y genotipos.

Su función es el desarrollo de procesos evolutivos en los que se crean y se analizan el comportamiento de un individuo o una población; mediante simulaciones virtuales que estudian la creación interacción y desarrollo de individuos en un entorno artificial. Para poder ejecutar esta tarea se sirve de un conjunto de instrucciones y acciones (AGs) que alteran el comportamiento, habilidades y genética de una población o un único individuo; con el objetivo de mejorar determinadas habilidades configuradas por el usuario previamente (función de evaluación, 2.7.4).

La meta de este programa es la evolución de un único individuo o conjunto para encontrar aquel genotipo o conjunto de genotipos que mejor satisfagan los criterios del usuario, hallar la región del espacio que maximice la función de evaluación. Para desempeñar esta tarea, Framsticks GUI consta de dos módulos interconectados:

- **Editor de genotipos:** encargado del diseño y creación de los individuos que van a ser objeto de estudio de cada experimento.
- **Simulador:** Evolución y análisis del comportamiento de los genotipos en un entorno artificial configurable. Para hacer un experimento se deben configurar un conjunto de parámetros relacionados con la simulación evolutiva, las características de los individuos, su función de evaluación, la apariencia del entorno artificial, el interfaz de simulación así como el resto de variables que se detallan en los siguientes apartados.

Para poder realizar un experimento es necesario utilizar los dos módulos de la plataforma, el editor para diseñar los modelos y el simulador para analizarlos.

⁴⁴ La GUI permite ejecutar todas las funciones disponibles en esta herramienta pero se recomienda al usuario el empleo del resto de módulos para determinadas acciones por ser más detallados y específicos para realizar dicha tarea.

En los experimentos se ejecutan distintas simulaciones de varios modelos de genotipos. Cada simulación se divide en dos grupos de genotipos: el conjunto de genotipos y la población temporal. Las instrucciones que guían y actúan sobre la simulación son los algoritmos genéticos, encargados de cumplir los objetivos del proceso evolutivo

El conjunto de genotipos es el grupo de individuos objeto de estudio sobre el que se aplican los AGs. Constituye las regiones del espacio que han sido seleccionadas como la más óptima solución al problema que plantea el usuario.

Para analizar el comportamiento de los individuos, hay que simularlos, la población temporal constituye el grupo de genotipos que está siendo simulado en un determinado instante del experimento. Mediante la simulación de su comportamiento se recogen el conjunto de parámetros que evalúan de forma individual a cada sujeto. Esta población está en continuo cambio, la cantidad de energía de cada uno determina el tiempo de simulación, si la energía es igual a cero, el genotipo habrá agotado su vida y será eliminado de la simulación.

El método de análisis es la función de evaluación, cuyo resultado es el valor de Ajuste. La función de evaluación se encarga de evaluar el comportamiento y las habilidades adquiridas por cada uno de los genotipos durante la simulación. Recoge todo el análisis y estudio de su comportamiento, la distancia recorrida, las colisiones que ha tenido, su velocidad máxima, todo ello resumido y jerarquizado por medio del valor de Ajuste⁴⁵. Los parámetros que participan en la función de evaluación son fijados por el usuario y el resultado obtenido se representa en relación con el resto de valores del grupo de genotipos (ranking).

El avance del experimento se monitoriza mediante la función de evaluación. Si el valor de Ajuste no ha mejorado en un período de tiempo determinado o respecto a un valor de referencia, se detiene la simulación y el experimento. El avance de la simulación puede ser detenido en cualquier instante y obtener una foto del estado de la simulación en un determinado momento; es adecuado cuando se realiza un experimento y se quieren observar las fases evolutivas por las que van pasando los distintos genotipos.

3.2 Módulos que forman Framsticks GUI

Framsticks GUI está formado por varias partes, cada una cumple una función determinada. Se dividen en dos tipos, las ventanas de edición y configuración, y las ventanas relacionadas con el análisis, simulación y estudio de las distintas partes y tareas que realiza el programa.

3.2.1 GUI como creador y editor de genotipos

Framsticks, es una plataforma de desarrollo compuesta por varios elementos y uno de ellos es la interfaz gráfica del mismo nombre. Esta herramienta tiene una función que permite la creación y edición de genotipos. El editor de genotipos de Framsticks permite la creación, edición, visualización y registro de un genotipo en un entorno artificial. A través de la GUI, es posible el desarrollo y diseño de genotipos que pueden ser simulados directamente. Tiene la ventaja de poder realizar, desde una misma aplicación, la edición y prueba de un modelo en un entorno artificial.

⁴⁵ En ocasiones se crean varios modelos idénticos de un mismo genotipo, por lo que su función de evaluación será la mejor de todas las instancias simuladas.

Es compatible con todos los tipos de codificaciones que se utilizan en Framsticks. Permite el diseño y creación de genotipos codificados en cualquiera de la sintaxis de Framsticks, al contrario que otras aplicaciones⁴⁶. Además de configurar los genotipos se puede ver de forma detallada las características de los modelos que han sido evolucionados o que han sido simulados; para analizar las transformaciones producidas por los AG y su interacción con él mismo y el resto de individuos.

Se puede observar la apariencia del individuo en el entorno artificial así como las conexiones de todas las partes, nexos y neuronas que lo forman⁴⁷. El eje de coordenadas es diferente al de FRED⁴⁸, en la siguiente figura, Figura 3.2-1, se muestra un esquema con los ejes de coordenadas por defecto de la GUI.

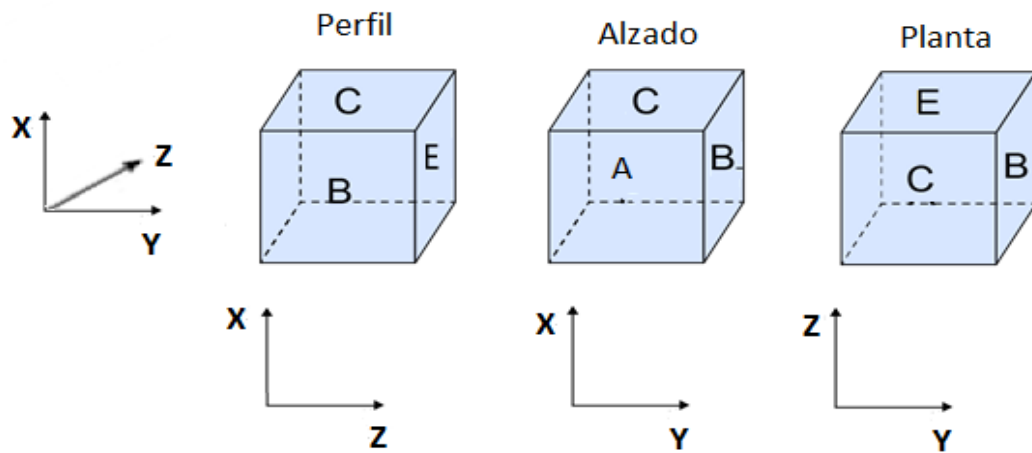


Figura 3.2-1 Ejes de Coordenadas Framsticks GUI

Al crear un genotipo en FRED y luego utilizarlo en la GUI, se representa abatido sobre el plano zy, si se quiere simular y se fija una posición y orientación inicial, el individuo no podrá avanzar porque su posición inicial será tumbado; para corregir este problema existen dos maneras, crear el genotipo siguiendo los ejes de la GUI o al configurar la simulación, fijar la orientación del genotipo en 180°. En la Figura 3.2-2 se explica gráficamente el abatimiento de los ejes.

⁴⁶ En la aplicación FRED, de esta misma herramienta, sólo se permite trabajar con genotipos que sean representados siguiendo la sintaxis de codificación f0. Para más información sobre esta aplicación, consultar el Capítulo 4.

⁴⁷ Para poder hacer un correcto y detallado estudio del comportamiento de un individuo en un entorno artificial de simulación se recomienda usar Framsticks Artificial Life(AL).

⁴⁸ Para más información sobre FRED, consultar el Capítulo 4.

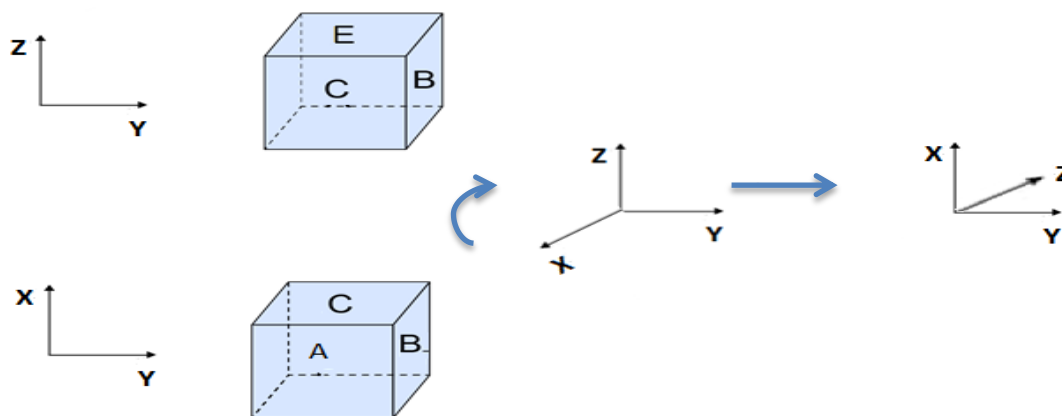


Figura 3.2-2 Cambio de ejes de la aplicación FRED a Framsticks GUI

En relación con el sistema de control del genotipo, se puede observar la estructura de la red neuronal y ver las conexiones y nexos a los que se une (Figura 3.2-3, (2)).

Respecto a otras aplicaciones de la plataforma, representa la información de manera más esquemática y resumida que en el caso de otros editores como FRED; que se centra más en los detalles de las partes de los individuos.

3.2.1.1 Partes del editor

Está formado por diversos apartados, relacionados⁴⁹ tanto con las características propias del genotipo, como su estado en la simulación, posición, energía, generación etc. La información respecto a la simulación, no alteran las variables propias del individuo y sólo muestran información sobre como evoluciona en el entorno artificial.

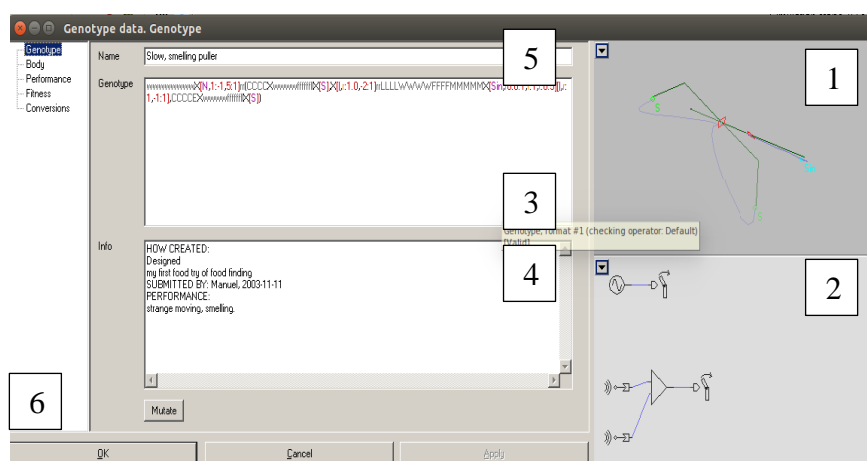


Figura 3.2-3 Editor de Framsticks GUI

⁴⁹ Por eso los módulos de Framsticks están interconectados, en el editor hay información relacionada con parámetros de la simulación y en la ventana de simulación se resumen las características de los genotipos.

En la Figura 3.2-3 aparecen los apartados que forman parte del editor de Framsticks; la ventana que muestra la apariencia del individuo en la simulación(1), la ventana (2) que representa la red neuronal del genotipo mediante símbolos que representan cada una de las neuronas y sus conexiones. En el apartado (3), se encuentra los parámetros, representado mediante cadenas de caracteres que caracterizan al individuo. En el apartado (4), se detalla la información relativa a la creación del genotipo y por último la opción (5) muestra el nombre asignado al individuo

En el menú de la izquierda (6), aparecen los campos configurables del genotipo. La primera opción de la derecha, *Genotype*, muestra las características del individuo, codificadas en un formato determinado.⁵⁰ El segundo campo, *Body*, se resumen las propiedades y las partes que forman el cuerpo del genotipo, el número de patas, nexos etc. *Performance*, engloba todos los parámetros relacionados con el comportamiento del genotipo en la simulación, su vida, la velocidad con la que se desplaza, la posición y un conjunto de campos configurables por el usuario.

En el apartado *Fitness*, se especifica el valor de la función de evaluación. El valor real obtenido mediante el cálculo de la función de evaluación y el valor final, que depende del comportamiento del resto de genotipos⁵¹. En el último campo, *Conversions*, se puede modificar el tipo de codificación de los distintos genotipos, siempre y cuando sean compatibles.

La sintaxis de codificación utilizada determina el tipo de evolución del genotipo. La utilidad de este apartado se basa en que como no todos los programas de la plataforma soportan todos los formatos de codificación mediante esta opción se puede pasar de un formato a otro, dentro de unos límites de compatibilidad.

3.2.2 GUI como simulador de procesos evolutivos

El estudio de los robots evolutivos se realiza mediante la ejecución de las simulaciones; que modifican su sistema de control y estructura física en función del comportamiento del individuo en un entorno artificial y atendiendo a unos objetivos que el usuario establece.

Este módulo es el encargado de la creación, ejecución y configuración de las simulaciones a través de un conjunto de parámetros. Las simulaciones son configurables, se pueden modificar las distintas acciones que ejecutan los scripts que las lanzan⁵². Existen simulaciones ya configuradas que permiten la evolución y análisis de distintos parámetros de la simulación⁵³, cada una de ellas provoca una evolución distinta sobre los individuos y analiza determinados parámetros.

La GUI sólo permite la ejecución de simulaciones creadas previamente, tanto por el usuario como las que proporciona la herramienta, para poder configurar los parámetros de la simulación, primero hay que cargar la simulación y luego ejecutarla. Los ejemplos de simulaciones⁵⁴ creados

⁵⁰ Framsticks GUI es compatible con todos los formatos de codificación de la plataforma.

⁵¹ Ranking Fitness

⁵² En la carpeta scripts, del directorio de Framsticks se encuentran los distintos scripts que gobiernan cada una de las simulaciones, alterando estos ficheros, se modifican las propiedades y el desarrollo de las distintas evoluciones.

⁵³ La información detallada sobre el funcionamiento y las características de cada uno de los ejemplos de simulaciones creados en la GUI aparece en el [4]

⁵⁴ En el caso de este TFG se utilizará el ejemplo de simulación standard

por la herramienta permiten el estudio de los genotipos siguiendo un modelo de simulación preestablecido.

Se pueden modificar los parámetros de la simulación pero el orden y prioridad de ejecución de cada una de las premisas está configurado previamente, así como el tipo de simulación. Por ejemplo en la simulación *reproduction*⁵⁵ la evolución se realiza a través de la reproducción asexual de los genotipos, *Neuroanalysis* estudia la señal de salida de las neuronas de los genotipos que participan en la simulación y realiza un estudio atendiendo a este parámetro.

3.2.2.1 Conocimientos previos

Para poder realizar el estudio lo primero que hay que saber es que los genotipos se dividen en dos grupos, el conjunto de genotipos donde se guarda toda la información sobre su actuación en el entorno artificial y su evaluación; y el grupo de genotipos que se simulan. Sobre el primer conjunto es en el que se aplica los distintos AGs que provocan la evolución de la especie y en el segundo grupo se evalúa su actuación en un entorno artificial.

La ejecución de los distintos AGs durante la simulación provoca la selección eliminación y reemplazo de los modelos que intervienen en la simulación, son los encargados de guiar la simulación hacia la solución que más se acerca a los criterios establecidos por el usuario.

3.2.2.2 Fases de la simulación

Las simulaciones evolutivas se dividen en tres fases principales: creación, simulación y evaluación de genotipos. A su vez estas tres fases se dividen en subfases, cada una afecta un grupo de genotipos y la secuencia y prioridad de estas fases la determinan los algoritmos genéticos. Las fases son las siguientes:

1. Creación de genotipos
 - a. Selección
 - b. Generación de descendencia
 - c. Eliminación
2. Simulación
 - a. Introducción de genotipos en la población temporal.
 - b. Recolección de los distintos parámetros que determinan el comportamiento de cada genotipo hasta el fin de energía de cada individuo.
3. Evaluación del conjunto de genotipos
 - a. Análisis y procesado del comportamiento
 - b. Asignación del valor de Ajuste mediante la función de evaluación
 - c. Vuelta al punto 2 hasta cumplimiento de criterio de terminación

⁵⁵ Para más información sobre las distintas evoluciones implementadas en Framsticks, consultar [3]

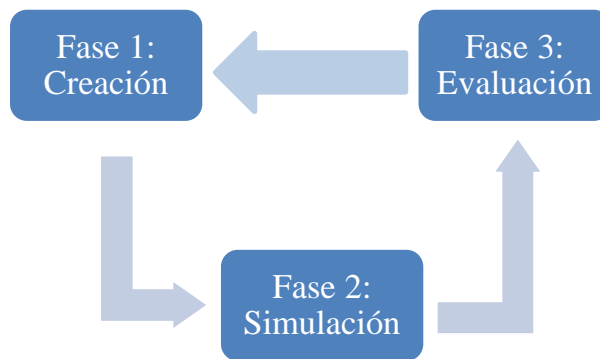


Figura 3.2-4 Fases de la simulación

El fin de la simulación se produce cuando se cumple el criterio de terminación, pero hasta que se produzca este momento, la simulación realiza un proceso cíclico, en la Figura 3.2-4 se representa un esquema del proceso.

En el siguiente apartado se muestran las distintas secciones donde se ejecutan las fases de la simulación.

Estas fases están ligadas al flujo de la simulación (Figura 2.8-1), son las acciones que se producen para pasar de un módulo a otro (Población temporal y Conjunto de genotipos). En la población temporal se crean los nuevos individuos (Fase 1), se observa su actuación (Fase 2: Simulación); se evalúa su comportamiento (Fase 3) y en función de los resultados son introducidos o no en el conjunto de genotipos.

3.2.2.3 Secciones de la interfaz de simulación

La interfaz de simulación, se divide en las siguientes secciones, Figura 3.2-5, cada una de ellas está relacionada con un conjunto de individuos y ejecuta una serie de acciones sobre ellos.

El conjunto de genotipos se recoge en la Figura 3.2-5, (5) y la población temporal en la Figura 3.2-5, (6); a su vez la población temporal puede tener dos clases de individuos distintos, los genotipos y la comida Figura 3.2-5, (2). Para conocer las características morfológicas y neuronales de las instancias de cada genotipo se utiliza la ventana Figura 3.2-5, (4). En esta ventana, se puede simular el comportamiento de cada una de las neuronas que forman el cerebro del individuo, mediante una función que permite la edición de la señal de entrada de las neuronas; con el objetivo de analizar el comportamiento individual de cada uno de los elementos que forman la red neuronal, para poder predecir el comportamiento del individuo. En la Figura 3.2-6, se encuentra representada la simulación del comportamiento de una neurona de dos genotipos simulados en la GUI.

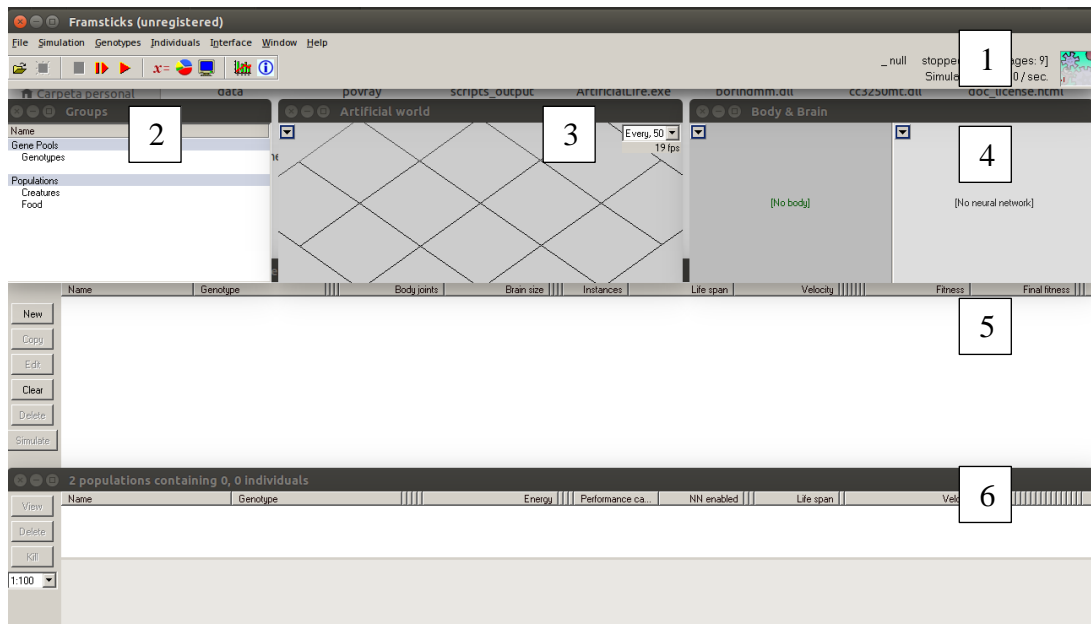


Figura 3.2-5 Secciones de la simulación⁵⁶

Las acciones que provocan la evolución de los individuos se ejecutan atendiendo a los parámetros de la simulación, para configurarlos existen dos opciones, a través de la GUI, Figura 3.2-5, (1) o modificando el código de los scripts que gobiernan sobre las distintas simulaciones.

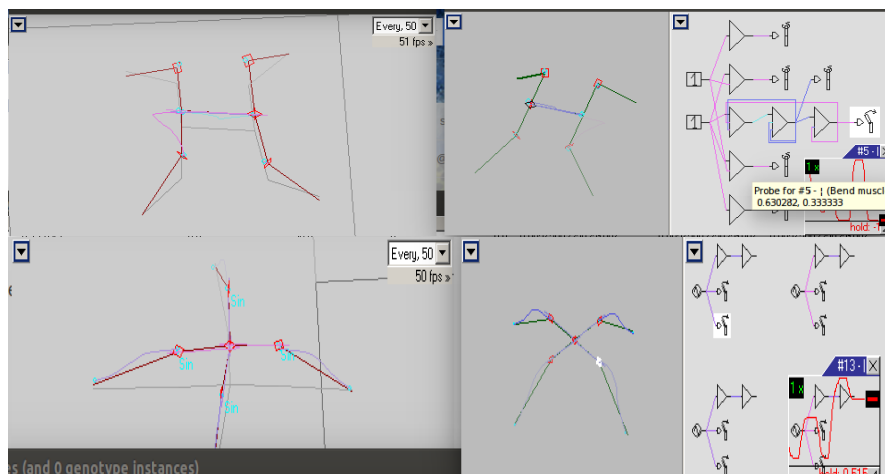


Figura 3.2-6 Ejemplos de los modelos analizados.⁵⁶

La relación de cada sección con los grupos y fases se detalla a continuación:

- Fase 1, Figura 3.2-5, (6): Se crean nuevos genotipos a partir de los existentes en el conjunto de genotipos o un gen inicial; los nuevos individuos son introducidos en la población temporal y simulados en un entorno artificial (3) de acorde a unos parámetros configurados en la simulación.
- Fase 2-3, Figura 3.2-5, (5): El análisis de su comportamiento es recogido y evaluado en el conjunto de genotipos (5).

⁵⁶ Imágenes obtenidas de la GUI.

3.2.2.4 Parámetros de la simulación

Conjunto de instrucciones que dirigen el desarrollo de la evolución y determinan sus propiedades, las capacidades de los conjuntos de genotipos, el número de individuos simulados etc. Existe una gran variedad de criterios, la cantidad de energía que se les asigna, el envejecimiento, el tiempo de estabilización del individuo⁵⁷, la situación inicial etc. Tanto las instrucciones (AGs) como parámetros y criterios que pueden gobernar la simulación, aparecen detallados en [4]; en este trabajo, para cada ensayo evolutivo se han configurado determinados parámetros y cada uno de ellos aparece resumido en las siguientes tablas resumen: Tabla 5.3-2, Tabla 5.3-5, Tabla 5.3-9 y Tabla 5.3-11.

Mediante estos parámetros se configura el objetivo de la evolución seleccionando los criterios y pesos que forman parte de la función de evaluación y que devuelven el valor de Ajuste de cada una de las instancias que se simulan. Los parámetros de los AGs que se van a ejecutar en la simulación se configuran también en este apartado.

La plataforma Framsticks posee dos formas de configuración de estos parámetros que se complementan. La modificación indirecta de los scripts que gobiernan la simulación o el uso de la GUI para la configuración. La Figura 3.2-7 y Figura 3.2-8 son capturas de pantalla de los botones y parámetros de la interfaz gráfica de la GUI.

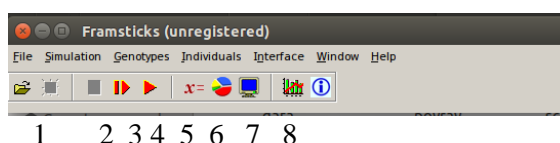


Figura 3.2-7 Botones de la GUI

Para poder modificar los parámetros desde la GUI, se dispone del botón 5, Figura 3.2-7, seleccionándolo aparece un menú desplegable con los distintos parámetros y opciones que se pueden configurar en la simulación a través de la GUI, en la Figura 3.2-8, se muestra la ventana que aparece con el conjunto de variables configurables.

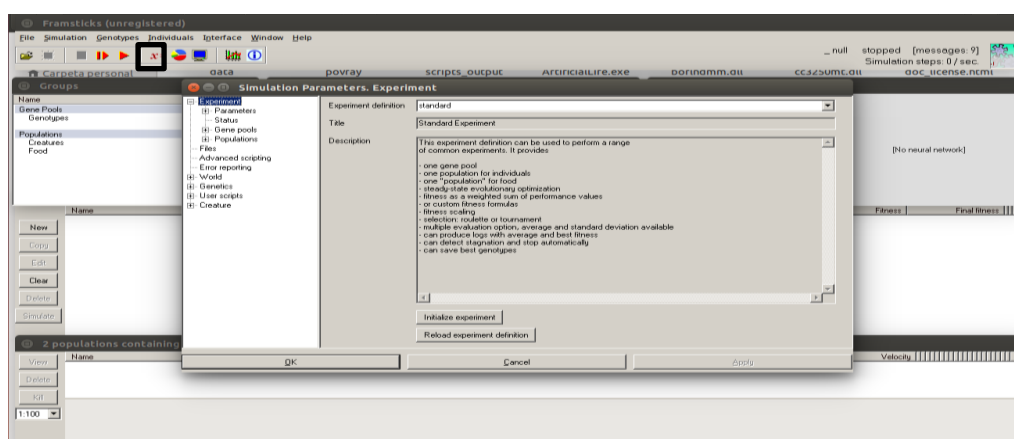


Figura 3.2-8 Parámetros de Simulación de la GUI

⁵⁷ El tiempo de estabilización del individuo aumenta la precisión del análisis, evitando medidas erróneas de la velocidad y la estabilización del genotipo; si la instancia simulada no se ha estabilizado, las medidas tomadas hasta ese momento no son válidas.

Una vez configurada la simulación, mediante los botones: (2), (3) y (4), se controla el avance de la simulación (Figura 3.2-7). En la siguiente imagen, Figura 3.2-9, se muestra unas capturas de pantalla de la simulación del comportamiento de dos modelos robóticos en un entorno artificial de la GUI:

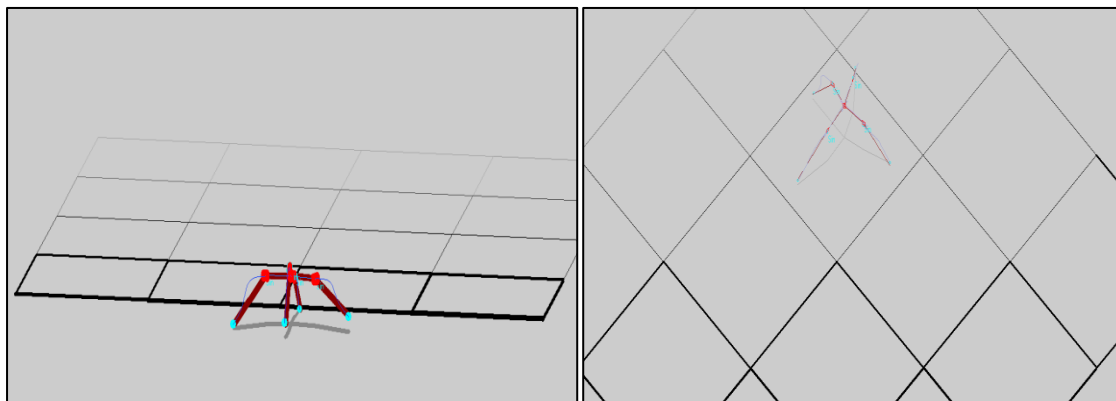


Figura 3.2-9 Capturas de pantalla de simulación en movimiento⁵⁸

3.2.2.5 Genética en la evolución

Para lanzar una simulación y configurar las partes de los genotipos que se quieren evolucionar, es necesario modificar la genética del individuo. Al igual que en la naturaleza, si se quisiera alterar las partes de un genotipo, se habría que modificar sus cromosomas, codificados en su ADN.

En Framsticks equivale a alterar el código de cada individuo; dependiendo del formato de código, la alteración y permutación de las partes del individuo varía.

Aunque todo lo explicado anteriormente influye en el desarrollo de la simulación y guía hacia el objetivo de la evolución, la codificación de los genotipos es muy importante. La sintaxis de codificación de los genotipos está fuertemente relacionada con la genética de la evolución, dependiendo de la forma de representación de los genotipos se limita parámetros concretos de evaluación. Si utilizamos el lenguaje de codificación f0 en vez de usar el f2, las características biológicas, del tipo asimilación, fotosíntesis, no se podrán evolucionar, ya que la sintaxis no lo permite.

Como en este trabajo, se quiere analizar el aumento de la velocidad de los genotipos en función de su estructura física, es decir la evolución del cuerpo del individuo para aumentar la velocidad manteniendo su cerebro intacto. La genética cobra mucha importancia, limitando el módulo de los robots que pueden modificar, permitiendo centrar el proceso evolutivo en la longitud de las patas o en los grados de movimiento de las articulaciones etc. Al igual que los parámetros de las simulaciones, los módulos y los pesos que se pueden evolucionar, la genética de cada individuo, aparecen en las siguientes tablas: Tabla 5.3 2, Tabla 5.3 5, Tabla 5.3 9 y Tabla 5.3 11.

⁵⁸ Imagen obtenida durante la ejecución de una simulación en Framsticks GUI.

3.3 Simulaciones lanzadas desde la terminal, Framsticks CLI: Command Line Interface

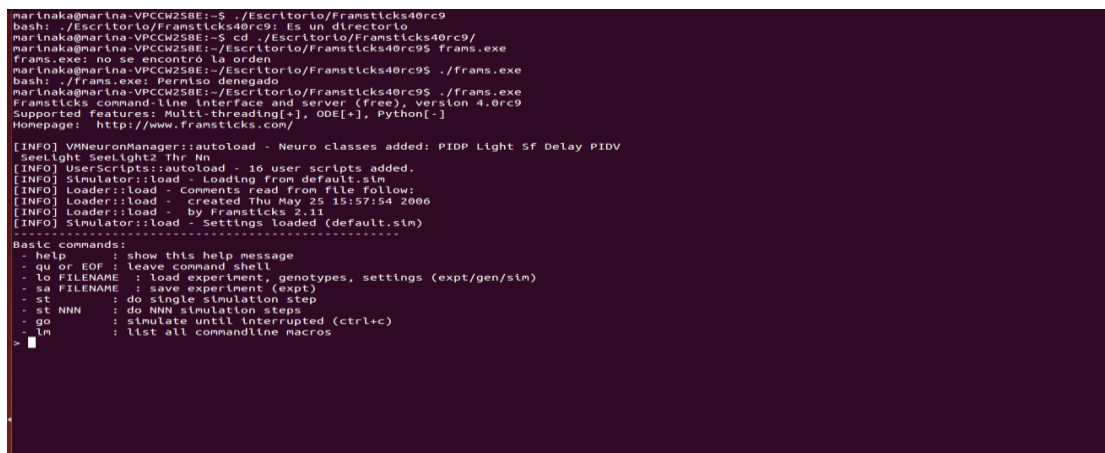
Además de la GUI, existe otra aplicación para la ejecución de simulaciones, Framsticks Command Line Interface (CLI). La Command Line Interface realiza las mismas funciones que la GUI pero se ejecuta mediante comandos lanzados desde la pantalla terminal.

Está diseñada para usuarios avanzados dado que hay que tener conocimientos de la sintaxis de los comandos y parámetros que se pueden configurar.

Mediante este programa se lanzan, configuran y analizan simulaciones a través de la terminal. Los resultados de estas simulaciones pueden ser guardados en un directorio especial y también se pueden cargar simulaciones configuradas previamente.

En los experimentos realizados en este TFG no se utiliza la CLI, porque se quiere ver el comportamiento de los genotipos simulados. Sin embargo si se quieren obtener los resultados la CLI es la aplicación ideal para esta tarea. Se obtienen más rápido porque no se visualizan y se pueden guardar todos los genotipos creados y el estado de la simulación en un instante concreto.

En la siguiente figura, Figura 3.3-1, se muestra una captura de pantalla de la terminal con la lista de comandos necesarios para ejecutar una simulación, los principales parámetros que hay que configurar para realizar los experimentos aparecen detallados en [4]



```
marinaka@marina-VPCCW258E:~$ ./Escriptorio/Framsticks40rc9
bash: ./Escriptorio/Framsticks40rc9: Es un directorio
marinaka@marina-VPCCW258E:~$ cd ./Escriptorio/Framsticks40rc9/
marinaka@marina-VPCCW258E:~/Escriptorio/Framsticks40rc9$ frams.exe
frams.exe: no se encontró la orden
marinaka@marina-VPCCW258E:~/Escriptorio/Framsticks40rc9$ ./frams.exe
bash: ./frams.exe: Permiso denegado
marinaka@marina-VPCCW258E:~/Escriptorio/Framsticks40rc9$ ./frams.exe
Framsticks command-line interface and server (free), version 4.0rc9
Supported features: Multi-threading[+], ODE[+], Python[-]
Homepage: http://www.framsticks.com/

[INFO] VMNeuronManager::autoload - Neuro classes added: PIDP Light SF Delay PIDV
Seelight Seelight2 Thr Nn
[INFO] UserScripts::autoload - 16 user scripts added.
[INFO] Simulator::load - Loading from default.sim
[INFO] Loader::load - Comments read from file follow:
[INFO] Loader::load - created Thu May 25 15:57:54 2006
[INFO] Loader::load - by Framsticks 2.11
[INFO] Simulator::load - Settings loaded (default.sim)
=====
Basic commands:
- help      : show this help message
- qu or EOF : leave command shell
- lo FILENAME : load experiment, genotypes, settings (expt/gen/sim)
- sa FILENAME : save experiment (expt)
- st        : do single simulation step
- st NNN    : do NNN simulation steps
- go        : simulate until interrupted (ctrl+c)
- lm        : list all commandline macros
```

Figura 3.3-1 Captura de pantalla de Framsticks CLI

3.4 Pasos para realizar una simulación

Los pasos que se deben seguir para realizar una simulación, son los siguientes:

1. Definir el objetivo de la simulación: dirige la simulación evolutiva a un objetivo determinado.
2. Seleccionar el tipo de simulación: depende del objetivo, si es directo, el tipo de simulación será directa. Si por lo contrario lo que queremos analizar es la supervivencia de los genotipos, la simulación será indirecta.

3. Modelo de genotipos: elegir el genotipo o conjunto de genotipos sobre los que queremos realizar el experimento.
4. Codificación de los genotipos: dependiendo del objetivo y el tipo de simulación, se elige la codificación que más se adapte a las necesidades del usuario o qué mejor cumpla los criterios fijados en los objetivos.
5. Selección de los AGs que van a guiar la simulación y a actuar sobre el genotipo o conjunto de genotipos de la simulación.
6. Fijar la capacidad máxima de los grupos de genotipos: población temporal y conjunto de genotipos y otros parámetros que influyen en la simulación⁵⁹.
7. Configuración de la genética
8. Lanzar la simulación
9. Analizar los resultados

⁵⁹ El resto de parámetros que se ha configurado para el desarrollo del experimento evolutivo de este trabajo aparecen en la Tabla 5.2-1

CAPÍTULO 4

FRED: Framsticks Editor

4.1 Definición y características

FRED, Framsticks Editor, es la aplicación encargada del diseño y creación de genotipos; es la interfaz gráfica desarrollada para diseñar y editar modelos robóticos. Los modelos robóticos son los genotipos que luego se utilizan en las simulaciones; todos están codificados usando la sintaxis más básica, f0.

A través de FRED, se puede diseñar desde la estructura física del genotipo hasta la configuración de la red neuronal. Está dividida en secciones, en función del tipo de configuración, directa o indirecta y del tipo de objeto que se va a configurar.

Es una aplicación diseñada para todo tipo de usuarios; gracias a sus ventanas de configuración directa, permite de forma rápida, sencilla y visual configurar los distintos objetos que conforman la estructura de un genotipo. De manera indirecta, muestra como es su código siguiendo la sintaxis f0, con todas sus propiedades detalladas para que los usuarios más avanzados puedan modificar el código y observar los resultados en una misma plataforma. En la Figura 4.1-1 aparece una captura de pantalla de la aplicación, con todas las secciones disponibles.

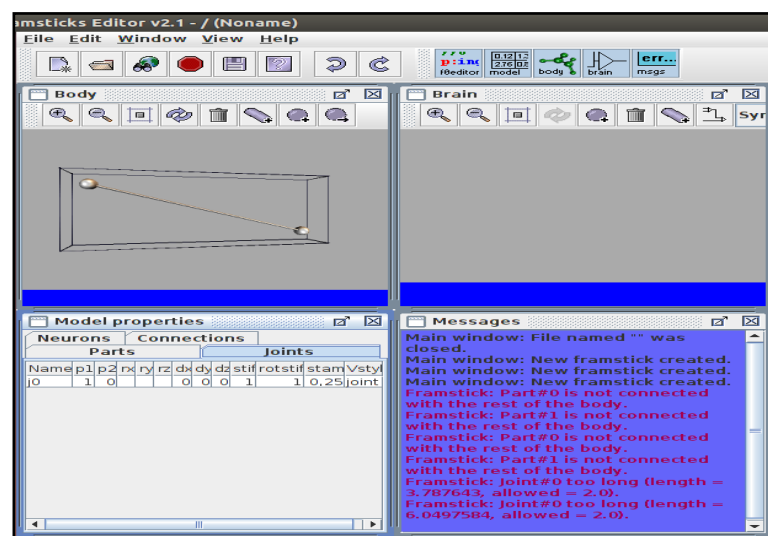


Figura 4.1-1 Fred Editor

Trabaja con un tipo de archivos especial para esta plataforma; sólo permite la edición y creación de archivos del tipo .gen. Los modelos que se crean se guardan por defecto en este formato para poder expórtalos al resto de aplicaciones.

Para utilizarla es necesario tener instalado en el ordenador un entorno de desarrollo de programas para escribir aplicaciones Java, debido a que Fred está implementado en Java.

4.2 Elementos de la Interfaz

Está compuesto por varias partes, dependiendo del tipo de objeto que se va a configurar y de si su configuración va a realizarse de manera directa o indirecta.

Para la configuración directa estan *Body* y *Brain*, ventanas de edición que representan visualmente las características del modelo. Gracias a estas dos ventanas de edición, el usuario puede crear y modificar pulsando directamente sobre el objeto, permitiendo la configuración de la estructura morfológica y neuronal fácilmente sin la necesidad de modificar el código del individuo.

Por otro lado, la ventana *Model properties*, muestra las propiedades de las distintas partes que forman el modelo mediante tablas. Esta última permite la configuración de los modelos de forma indirecta a través de la modificación de su código. La consola de mensajes es el último elemento de la interfaz y es la encargada de mostrar los cambios realizados por el usuario sobre los genotipos.

4.2.1 Menú principal

El menú principal está formado por un conjunto de botones que realizan acciones sobre los archivos que contienen los distintos genotipos. En la Figura 4.2-1, se encuentra una captura de pantalla de los botones del menú de FRED.



Figura 4.2-1 Botones del menú principal

Se pueden crear, cerrar y guardar diferentes genotipos, así como abrir modelos desde archivos locales o descargarlos de la base de datos de Internet de Framsticks. Otorga el control de las acciones ejecutadas, pudiendo deshacer y rehacer los cambios realizados. También es el encargado de la visualización de las partes que conforman la herramienta. Los 5 botones situados a la derecha del menú (Figura 4.2-1) representan cada una de las partes de la aplicación, pudiendo el usuario mostrar o esconder los distintos elementos. Tiene un botón de ayuda que despliega un resumen de las características y partes de la aplicación para ayudar al usuario en el manejo de la herramienta.

4.2.2 Ventana *Body*

Mediante esta ventana se puede modificar directamente la estructura morfológica de los genotipos, sin sobrecribir el código. En ella se representa la disposición de cada uno de los nexos y las articulaciones del modelo; también incluye las propiedades de cada uno de ellos, la dirección del desplazamiento, las neuronas a las que está unidas etc. En la Figura 4.2-2 aparecen varios ejemplos de genotipos representados en la ventana de *Body*; como se puede apreciar en los modelos de los extremos, en esta ventana también aparece las neuronas que están unidas a cada elemento de la estructura y la dirección del movimiento que puede realizar el genotipo.

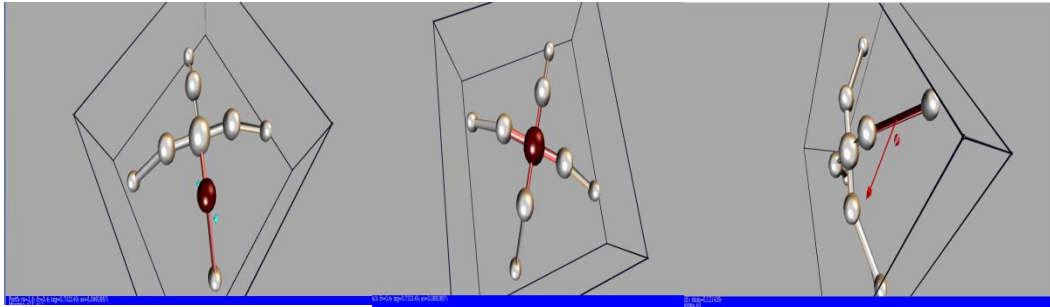


Figura 4.2-2 Ejemplo de varios genotipos representados en la ventana *Body*

En la Figura 4.2-3, los botones de la parte superior de la ventana muestran las distintas acciones que el usuario puede realizar. Son los responsables del diseño y visualización de la estructura morfológica del modelo robótico. En el centro de la Figura 4.2-3, en la que aparece la ventana *Body*, se muestra un rectángulo tridimensional que delimita el espacio que va a ocupar el genotipo. El tamaño del cuadro va aumentando a medida que se le añaden elementos al modelo.

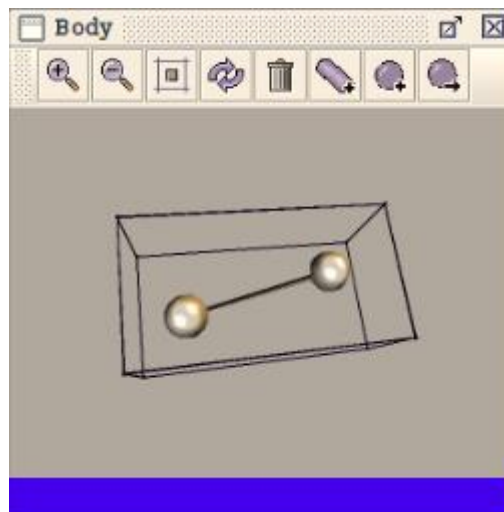


Figura 4.2-3 Vista de la ventana *Body*

Por defecto los ejes de coordenadas están configurados siguiendo el siguiente esquema de la Figura 4.2-4, no coinciden con los ejes de coordenadas utilizados en la GUI, en la Figura 3.2-2, se encuentra representada la diferencia entre ambos:

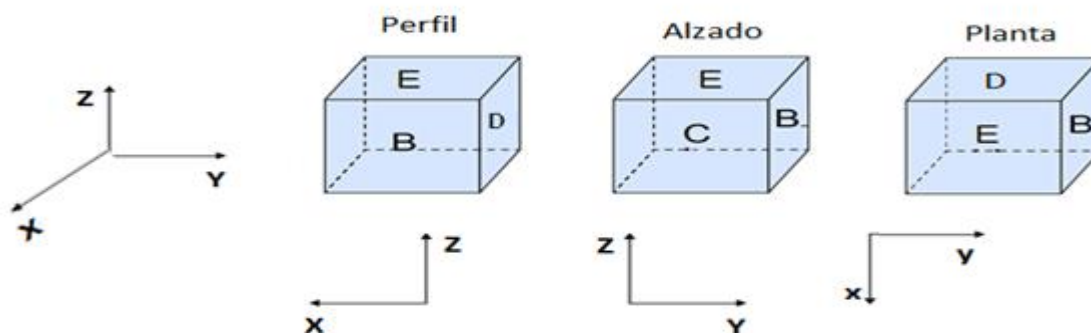


Figura 4.2-4 Ejes de coordenadas⁶⁰ en FRED

Si se mantiene seleccionado cualquiera de las partes del genotipo y se desliza el ratón se modifica su orientación. Mediante el ratón se puede configurar la localización en el plano de los distintos elementos; primero hay que elegir el plano y a continuación se selecciona la articulación o nexo y se arrastra hasta la posición deseada.

En la parte inferior de la ventana (Figura 4.2-3), se detallan las características de los elementos, para poder visualizar esta información basta con seleccionar la articulación o nexo y aparecen sus propiedades. En la Figura 4.2-5, se encuentran enumerados todos los botones disponibles en la ventana *Body*:

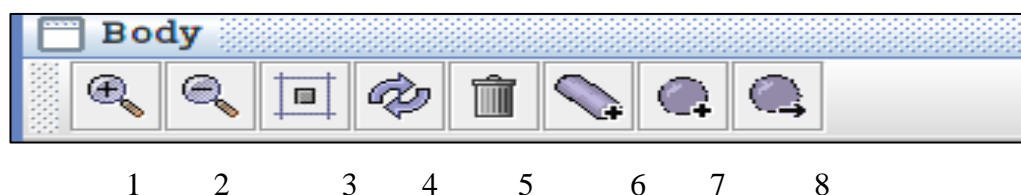


Figura 4.2-5 Botones de *Body*

Los botones 1 y 2 se encargan de acercar o alejar el genotipo, aunque esta función se puede llevar a cabo también con la rueda del ratón del ordenador. El botón 3 muestra el genotipo en la posición inicial por defecto. Con el botón 4, podemos rotar en los ejes x, y, z para poder ver la estructura del genotipo desde todos los ángulos.

El resto de botones son los responsables de la creación, alteración y eliminación de las articulaciones y nexos del sujeto. Con los botones 7 y 8 se añaden y mueven las distintas articulaciones, el botón 6 es para unir los nexos con las articulaciones, y el botón 5 sirve para eliminarlos.

Para crear la estructura morfológica de un genotipo, se han de añadir en primer lugar las articulaciones (botón 7) y después unir las mediante los nexos (botón 6). Para poder introducir una nueva articulación, hay que elegir primero el plano y luego seleccionar con el ratón el sitio

⁶⁰ En la aplicación Fred, Planta equivale a *FRONT*, Perfil a *PLANT* y Alzado a *SIDE*

donde se va a ubicar. La primera que se crea es situada por defecto en la posición (0, 0,0) y se denomina *Part0*, a continuación se van nombrando sucesivamente.

Si se desea modificar la posición de una articulación, se utiliza el botón 8, seleccionándola y arrastrándola con el ratón, se modifica su posición. Los nexos se crean mediante la unión de dos articulaciones (botón 6); hay que tener cuidado con esta función ya que Framsticks tiene delimitada la longitud de los nexos en 2 unidades, en el caso de que la longitud sea superior a dos, el genotipo será inválido y no podrá utilizarse en posteriores simulaciones.

Para modificar las propiedades de los distintos elementos, se selecciona y se pulsa con el botón izquierdo del ratón; desplegándose un menú con el que modificar sus características geométricas y relacionadas con la naturaleza del elemento (Figura 4.2-6).

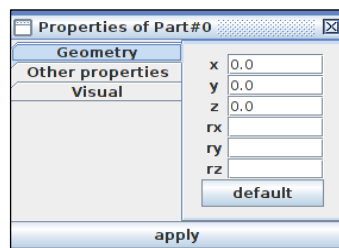


Figura 4.2-6 Ventana de propiedades

4.2.3 Ventana *Brain*

Brain, es la otra ventana de edición de configuración directa; que permite al usuario configurar la red neuronal del genotipo, desde el tipo de neurona que va a poseer, sus propiedades, pesos, conexiones con otras neuronas y la localización de cada neurona en la estructura física. En la Figura 4.2-7 aparece una captura de pantalla de la ventana *Brain* en la que se representa el cerebro de uno de los genotipos.

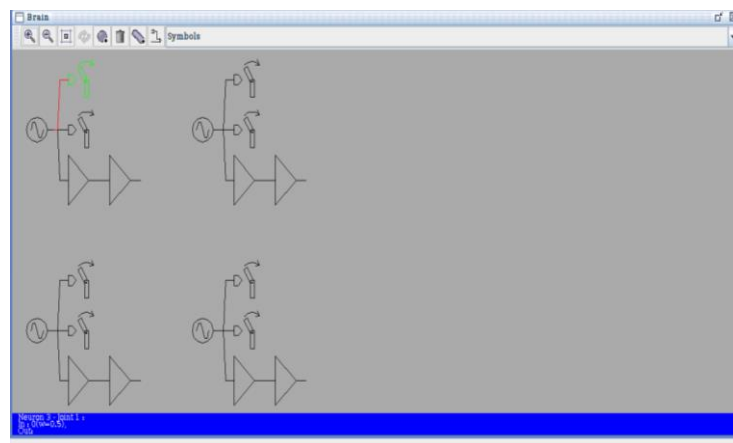


Figura 4.2-7 Ventana *Brain* con un genotipo de ejemplo

En la Figura 4.2-8, al igual que en la ventana *Body*, se encuentran numerados los botones de la ventana *Brain*. Los botones 1, 2, 3, 6 son iguales a los de *Body*, con el botón 5 se añaden distintos tipos de neuronas y con el 7 se crean las conexiones entre ellas. Se puede modificar la

alineación de las neuronas con la pestaña 8 y con la 9 se cambia la representación simbólica de las mismas.

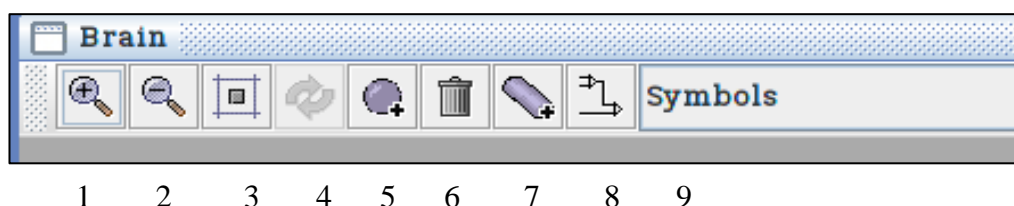


Figura 4.2-8 Botones de *Brain*

Del mismo modo que en el caso anterior, si seleccionas una neurona, en la parte inferior de la ventana aparece información relativa a ella, el número con el que se la identifica y sus conexiones. Otra similitud es la siguiente, si se quiere modificar las propiedades de las neuronas, seleccionando la neurona elegida y pulsando con el botón izquierdo del ratón, se despliega el menú de las propiedades, pudiendo modificar las características de cada una de las neuronas. También incluye las conexiones que tiene cada una de las variables, en el caso de que una de las conexiones fuera incompatible, la aplicación pondría el valor -1 por defecto y si el usuario modificase ese valor, la conexión no sería aplicada por ser incompatible con la configuración de la aplicación.

4.2.4 Ventana *Model properties*

La función de esta ventana es proporcionar al usuario una vista rápida de como está configurados los distintos elementos del genotipo. Está dividido por pestañas y cada una posee una tabla con las características de cada elemento, para modificarlas hay que escribir los nuevos valores directamente sobre la tabla y pulsar Intro.

Si se desea comprobar que se han modificado, en las ventanas Body y Brain aparecen los distintos elementos con los cambios realizados por el usuario. La Figura 4.2-9 aparece un ejemplo de la ventana Model properties.

Model properties														
Parts		Joints		Neurons		Connections								
Name	x	y	z	m	s	vol	fr	ing	as	rx	ry	rz	Vstyle	dn
p0	0	0	0	4	1	1	0,4	0,732	0,089				part	1
p1	-0,866	-0,5	0	2	1	1	0,4	0,732	0,089				part	1
p2	-2,068	-1,195	-0,905	1	1	1	0,4	0,636	0,121				part	1
p3	0,5	-0,866	0	2	1	1	0,4	0,732	0,089				part	1
p4	1,194	-2,068	-0,905	1	1	1	0,4	0,636	0,121				part	1
p5	0,866	0,5	0	2	1	1	0,4	0,732	0,089				part	1
p6	1,903	1,098	-1,145	1	1	1	0,4	0,636	0,121				part	1
p7	-0,5	0,866	0	2	1	1	0,4	0,732	0,089				part	1
p8	-1,193	2,069	-0,905	1	1	1	0,4	0,636	0,121				part	1

Figura 4.2-9 Model properties

4.2.5 Consola de mensajes

Es la parte de FRED encargada de mostrar los avisos del programa cuando el usuario realiza acciones que no son válidas como por ejemplo la limitación de la longitud de los nexos. En la Figura 4.2-10 aparece un ejemplo de estos mensajes:

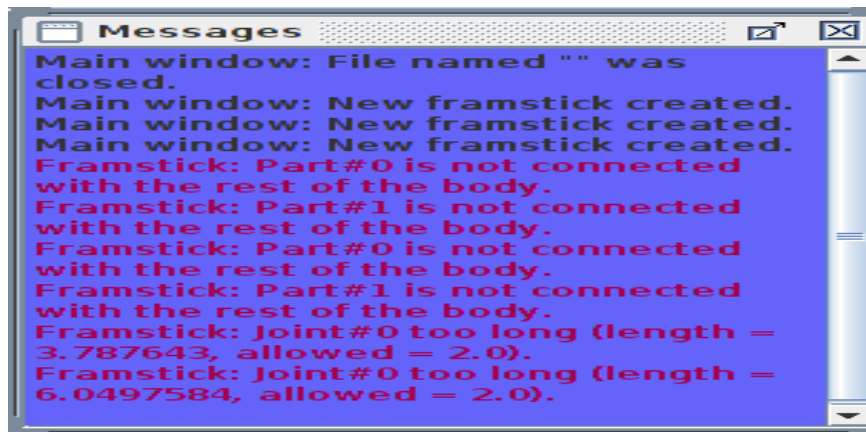


Figura 4.2-10 Consola de mensajes

Capítulo 5

Diseño y simulación del modelo

5.1 Introducción

En este apartado se va a realizar un ensayo evolutivo de un modelo robótico con la intención de conseguir optimizar su velocidad y distancia recorrida.

El primer paso es la creación del modelo robótico, como se mencionaba en la Introducción 1.1, se intenta conseguir un diseño que imite el desplazamiento animal, que realice un movimiento “natural”. Para ello, se hace un estudio de la locomoción del futuro robot y se presentan dos soluciones: un modelo cuadrúpedo que se desplaza realizando un movimiento similar al de una “tortuga” y otra versión cuadrúpeda que imita al movimiento de una “araña”. Ambas versiones son implementadas en Framsticks y se detallan en los siguientes apartados (5.2.1 y 0).

Una vez creado el modelo robótico, el siguiente paso es la selección de las pruebas para evolucionar el prototipo. Con estos experimentos se pretende también realizar un estudio de la influencia de las distintas estrategias evolutivas en los modelos; lo que implica que en las pruebas realizadas se aplican distintos criterios evolutivos y algoritmos genéticos. En la Figura 5.1-1 se muestra una clasificación de las pruebas en función de los objetivos y técnicas de cada experimento.

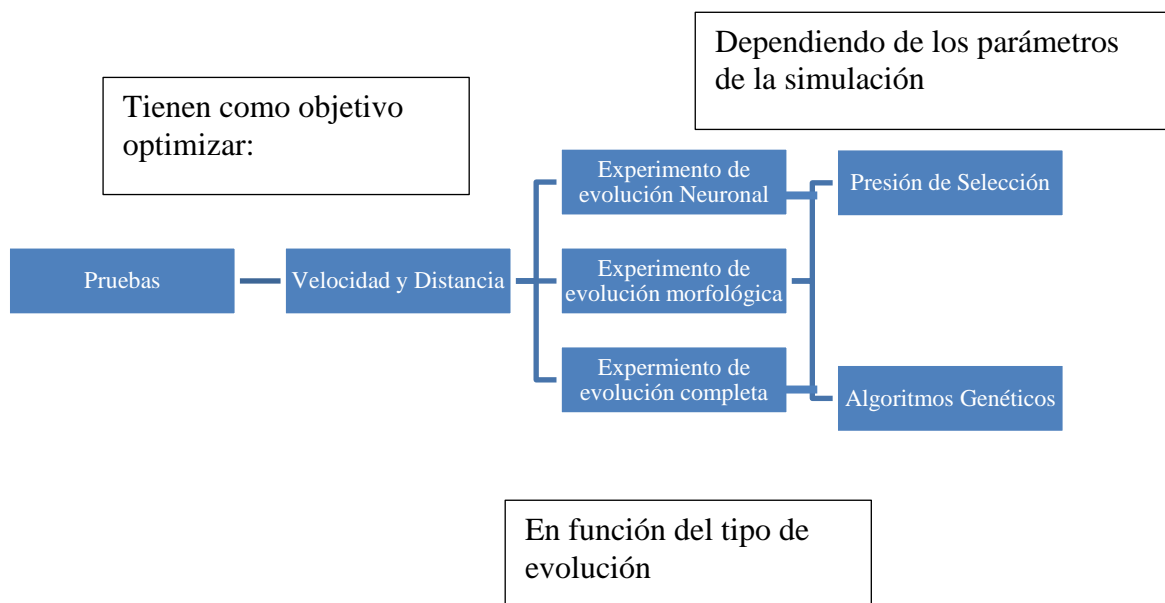


Figura 5.1-1 Clasificación de las pruebas

Los experimentos de evolución morfológica son complementarios a los de evolución neuronal. Primero se evoluciona el cerebro de los individuos para que la distribución de los distintos pesos y conexiones favorezca el desplazamiento. Y a continuación, esa configuración es sometida a pruebas evolutivas que modifican su estructura física.

Por tanto el orden de desarrollo de los experimentos⁶¹ cobra importancia y se realiza en función de las necesidades de trabajo. La secuencia de los experimentos tiene el siguiente orden:

1. Diseño modelo.
2. Pruebas de evolución neuronal.
3. Pruebas de evolución morfológica.
4. Pruebas de evolución completa.

Las pruebas realizadas siguen distintas técnicas que intervienen sobre la evolución; pueden afectar a la genética, limitando los elementos del modelo que pueden evolucionar, pueden aplicar distintos criterios de selección restrictivos, también elegir el tipo de reproducción de los individuos para generar descendencia, intentar evitar la aparición de mínimos locales y el estancamiento de la evolución etc.

Las variables que participan en cada una de las pruebas y los modelos iniciales se detallan en cada uno de los subapartados de los experimentos.

5.2 *Diseño del modelo*

El objetivo del trabajo es analizar las transformaciones producidas en los genotipos que aumentan su velocidad de desplazamiento. Es necesario construir un modelo que cumpla con una serie de requerimientos; por ejemplo, que sea capaz de moverse por sí sólo, que tenga autonomía, estabilidad, movimientos que favorezcan un desplazamiento dinámico y funcional etc.

La elección de los distintos elementos que conforman los modelos se debe al estudio del comportamiento de un conjunto de individuos con diferentes elementos simulados en el programa AL⁶²; debido a la naturaleza de las neuronas y las peculiaridades del entorno, este estudio ha surgido del análisis causa-error de los distintos modelos diseñados.

Se ha utilizado el programa GUI, Figura 3.2-6, para analizar el comportamiento de cada neurona de forma individual y ver cómo afecta al movimiento global, ha sido la única aproximación del movimiento, junto con Artificial Life⁶², que se ha empleado. El resto del estudio se ha desarrollado diseñando y probando distintos modelos hasta conseguir el prototipo utilizado.

⁶¹ En el caso de los ensayos de evolución completa no siguen ninguna secuencia de ejecución son independientes del resto de pruebas.

⁶² AL, Artificial Life, es el simulador gráfico 3D del comportamiento de los modelos robóticos en un entorno artificial, se detalla en el apartado 5.5.

La complejidad de estos experimentos se basa en el diseño y elección del modelo robótico. Imitar un determinado comportamiento utilizando los elementos disponibles de la plataforma guarda una gran dificultad, debido a las restricciones de los modelos y los tipos de neuronas disponibles en la plataforma.

Como afirmaba Karl Sims, la dificultad de un modelo dinámico se basa en la diferencia entre elegir que el robot haga un movimiento deseado o que su movimiento sea plausible. La complejidad radica en el diseño de su sistema de control y morfología de cara a un objetivo [1]. Por ejemplo, si se quiere que el genotipo realice un determinado movimiento, programarlo para que gire una extremidad 180° grados, el movimiento global del individuo puede no ser el adecuado; si por el contrario, queremos dotar al genotipo de un movimiento “natural”, puede que no ejecute acciones en la dirección que se haya elegido.

Por eso es mejor proveer al genotipo de pares de fuerzas y músculos necesarios para que realice un movimiento plausible, aunque se pierda el control sobre los movimientos exactos que realice.

5.2.1 Modelo con cuatro extremidades

La meta de este modelo es emular el movimiento de un animal cuadrúpedo. Se ha estudiado la locomoción de una tortuga para imitar su desplazamiento. Se pretendía dotar al modelo de sensores balanceadores que equilibraran al robot, sensores infrarrojos para evitar colisiones etc. Teniendo en cuenta las restricciones de la herramienta⁶³ el uso de estos sensores no ha sido posible y se han tenido que buscar alternativas de diseño basadas en los elementos disponibles en Framsticks.

Se ha elaborado una plantilla con palos magnéticos y se han estudiado los distintos elementos que debía poseer para poder desplazarse en un entorno artificial, en las siguientes figuras, se muestra el modelo de pruebas que se ha construido representado en la herramienta (Figura 5.2-1) y las versiones de plastilina (Figura 5.2-2 y Figura 5.2-3).

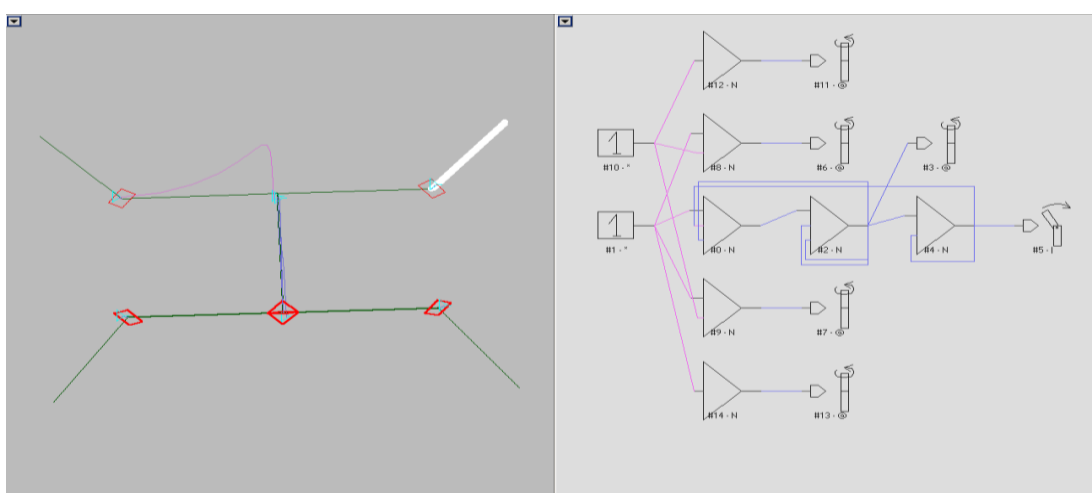


Figura 5.2-1 Esquema de FLINK representado en Framsticks GUI

⁶³ Para más información sobre las restricciones de la herramienta consultar en los distintos tipos de neuronas nexos y articulaciones disponibles para crear los robots. [4] y [16]

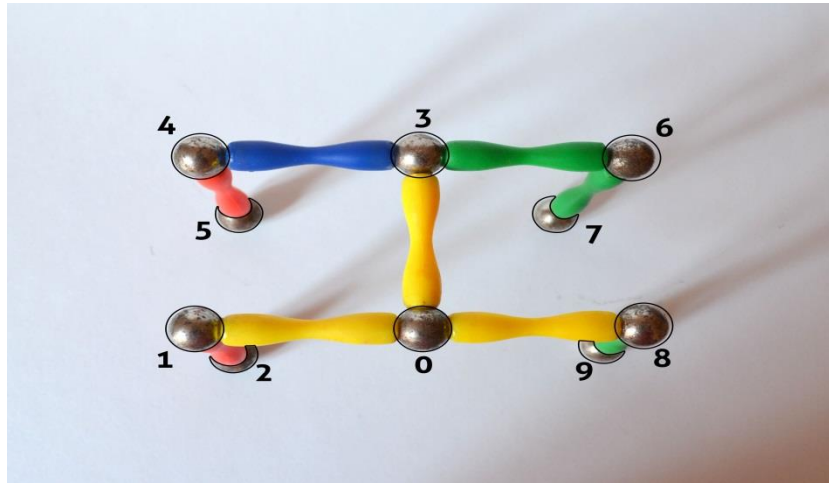


Figura 5.2-2 Planta de FLINK

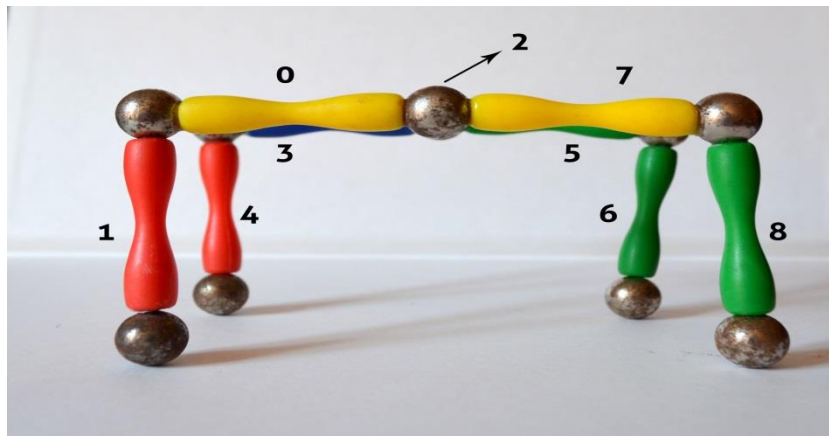


Figura 5.2-3 Alzado de FLINK

El modelo posee músculos rotatorios en los nexos 1, 4, 8 y 6 que favorecen el desplazamiento hacia delante, así como un músculo doblador en el tronco del modelo que permita el avance del genotipo.

Para aportar autonomía al individuo se han incluido dos neuronas con una salida constante igual a 1, cada una de ellas unida a una neurona sigmoidea; que a su vez se enlaza con el músculo que actúa sobre cada uno de los nexos de las extremidades.

El tronco está unido a una de las neuronas con salida constante y posee: un músculo giratorio, otro doblador y dos neuronas más cada una unida a cada músculo. En la Figura 5.2-4 se muestra la configuración inicial del cerebro del genotipo, bautizado como *FLINK*.

Cómo no ha sido posible el uso de balanceadores, el robot no compensa el peso cuando una de las extremidades se ha elevado, la locomoción de FLINK no equilibra el avance del robot ni mantiene fija las extremidades al detectar la elevación de las del lado opuesto. Su movimiento depende de los impulsos que envían las neuronas constantes y la salida de las neuronas sigmoideas que transforman la señal de entrada y transfieren la señal de salida a los músculos.

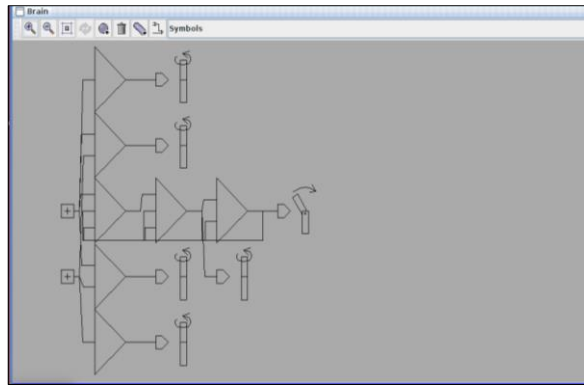


Figura 5.2-4 Cerebro de FLINK

La ventaja de este modelo es la sencillez de su estructura y su red neuronal, posee sólo cuatro patas y realiza un único movimiento de avance. Su cerebro está gobernado por dos neuronas constantes que transmiten impulsos al resto de neuronas, estas neuronas transforman los impulsos en señales que se transmiten a los músculos del genotipo provocando su movimiento.

El inconveniente de este modelo, de cara al experimento, es el escaso margen de operación. Al ser un modelo muy simple y provocar evoluciones limitadas en número y tipo de nexos, articulaciones y neuronas, provocan la limitación de la evolución.

5.2.2 Modelo con cuatro extremidades arácnido

Con el siguiente prototipo se pretende imitar el movimiento arácnido. La estructura del modelo está centrada en una sola articulación, unida al resto de extremidades. En la Figura 5.2-5 y Figura 5.2-6, se representan dos modelos con una estructura morfológica idéntica pero con distinta configuración neuronal.

En la versión 1, Figura 5.2-5, el sistema de control se basa en cuatro neuronas senoidales situadas en las articulaciones que se unen con los nexos de las extremidades. Cada una de estas neuronas está conectada a los músculos de su extremidad correspondiente. Sin embargo, en la versión 2, Figura 5.2-6, la red neuronal posee una única neurona senoidal ubicada en la articulación central y conectada con los músculos de todas las extremidades.

Con estos dos modelos se analizó cual era la configuración de la red neuronal que más favorecería el movimiento. En la primera versión, el objetivo era conseguir el desplazamiento del modelo variando la frecuencia de las señales senoidales. En la segunda versión se ha utilizado una sola neurona que gobierne todas las extremidades y para compensar el movimiento se han colocado neuronas de retardo en las extremidades.

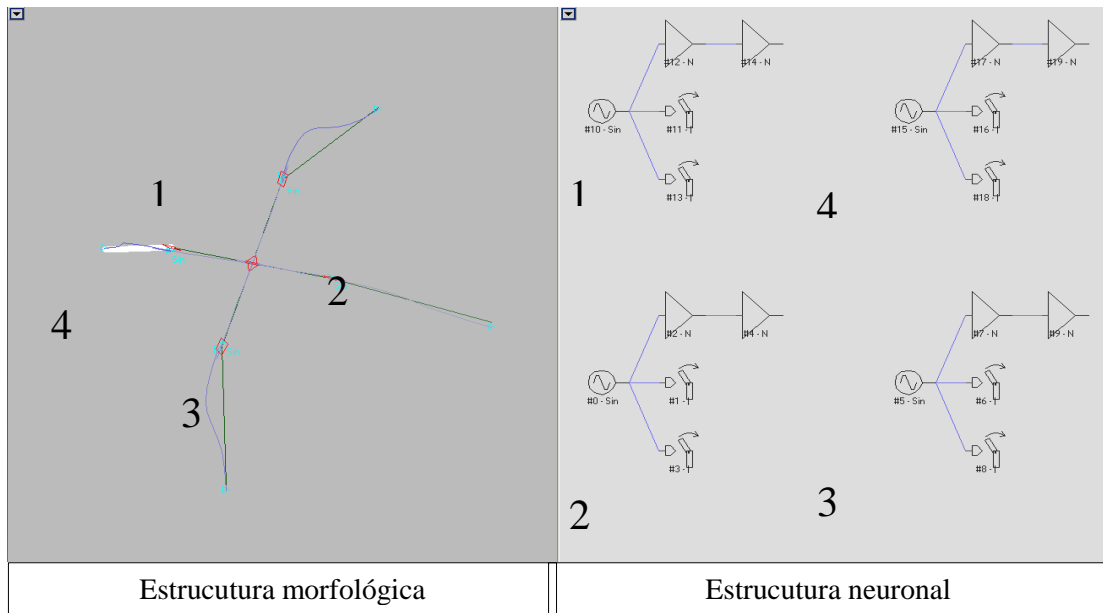


Figura 5.2-5 Modelo estilo araña versión 1

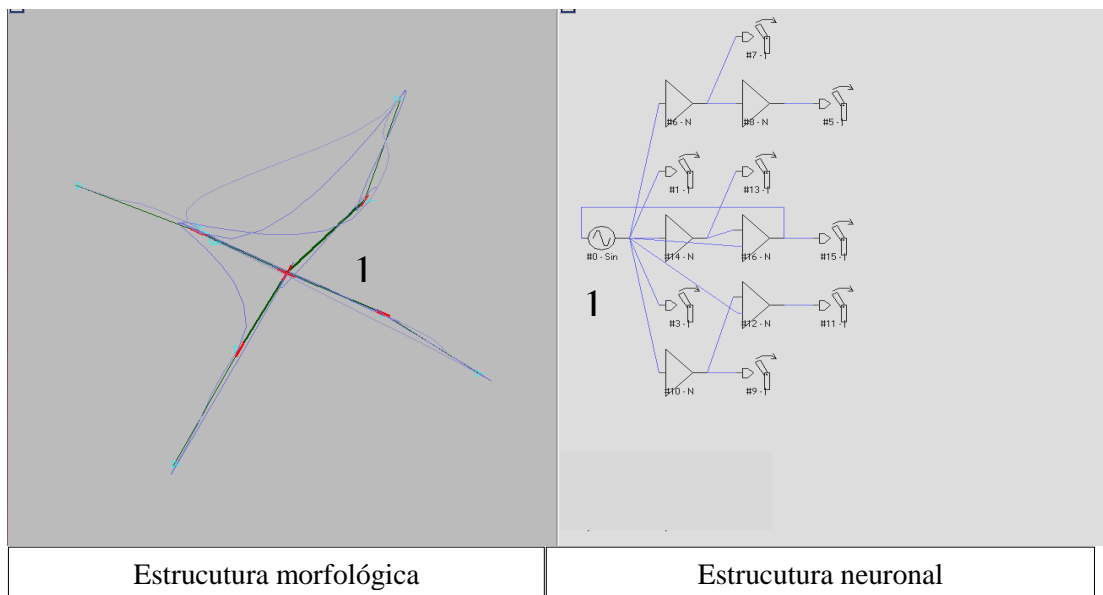


Figura 5.2-6 Modelo estilo araña versión 2

El inconveniente de estos modelos ha aparecido al diseñar un individuo que se desplace de manera funcional con la distribución explicada anteriormente. Como posee neuronas sigmoideas, no todas envían impulsos con la misma intensidad y el mismo signo, debido a la señal que reciben; provocando que los movimientos de las patas no se desplacen a la vez ni con el mismo ritmo ni con la misma dirección.

Aunque los modelos arácnidos son más complejos que el modelo anterior, se ha evaluado su comportamiento⁶⁴ y no cumplen los objetivos establecidos, por lo que ambas versiones no forman parte del experimento evolutivo.

⁶⁴ Las simulaciones del comportamiento de ambos robots se encuentran en [11].

5.2.3 Pruebas

Elegido el prototipo robótico, el siguiente paso es el desarrollo de las distintas pruebas que componen el ensayo. Para cada prueba se han lanzado varias simulaciones con distintos parámetros configurados en cada una de ellas. Sin embargo el tipo de simulación se ha mantenido constante, en todos los experimentos se ha utilizado la simulación estándar definida en la plataforma⁶⁵.

Las pruebas han generado distintas simulaciones que dependen de un conjunto de parámetros. Estos se clasifican en función de las técnicas evolutivas empleadas, distinguiendo entre técnicas que limitan la capacidad de los distintos grupos de genotipos (capacidad del conjunto de genotipos y número máximo de criaturas simuladas), otras que determinan la selección y eliminación de los distintos individuos del conjunto de genotipos (AG) y las que determinan el mecanismo de generación de descendencia (OG). Por último también se tienen en cuenta una serie de variables que condicionan el avance de la simulación y los módulos que pueden ser evolucionados (Genética). En el apartado 5.3 se encuentra la tabla resumen con los parámetros que participan en cada simulación.

A continuación se detallan el conjunto de pruebas realizadas así como el orden de ejecución de las mismas. Mediante los experimentos se pretende encontrar la configuración física y neuronal que aumente la velocidad de desplazamiento del prototipo.

A través de ellas también se quiere analizar la diferencia entre la evolución por partes y la evolución completa y como afecta distintas técnicas de evolución al objetivo.

Los resultados obtenidos se han clasificado en función de la máxima velocidad y distancia recorrida por cada una de las instancias y para poder analizar los cambios en la estructura física de los genotipos se ha utilizado una hoja de cálculo; donde se calcula la distancia entre cada una de las articulaciones, obteniendo la longitud de los distintos nexos (extremidades) que poseen los mejores genotipos de cada experimento. Se han comparado con el genotipo inicial y se han analizado los cambios producidos en la estructura.

El criterio de terminación de las pruebas ha sido el estancamiento, es decir, cuando en la evolución no se producía ninguna mejora en un tiempo determinado; se detenía, asumiendo que si transcurrido ese tiempo concreto no se conseguía ningún genotipo mejor el experimento había finalizado.

Todas estas simulaciones han sido realizadas a través de la GUI de Framsticks aunque también es posible mediante CLI.

5.2.4 Aspectos importantes de las simulaciones

Al iniciar la simulación de un genotipo, su posición inicial puede jugar un papel decisivo; dependiendo de las características del mundo artificial, si existe una zona rocosa o deslizante y determinados genotipos son situados allí, jugarán con desventaja debido a las características del terreno que hacen más difícil su comportamiento. Para evitar estos problemas hay dos opciones, configurar un mundo sin obstáculos o situar a los genotipos en la misma posición y orientación para que ninguno tenga ventaja sobre el resto.

⁶⁵ El nombre de la simulación utilizada es *Standad* y se encuentra disponible en la herramienta

Otro de los problemas que aparece al comenzar la simulación, es el momento en el que se empieza a analizar el comportamiento de los genotipos, es decir, el instante en el que empieza, por ejemplo, a recopilarse la distancia recorrida.

Esta situación supone un inconveniente importante a la hora de obtener los datos más exactos y verídicos posibles, cuando se empieza a simular un genotipo existe un período de tiempo hasta que se estabiliza estructural y neurológicamente. En este período⁶⁶ puede suceder que el genotipo se desplace con la inercia de su posición inicial, la simulación empiece a parametrizar este movimiento y a continuación se desestabilice.

El comportamiento durante este espacio de tiempo no sería válido ya que es aleatorio y puede estar sometido a vibraciones iniciales, por lo que no debe de introducirse en la función de evaluación ni tenerse en cuenta en los distintos criterios establecidos.

También se han de predecir diversas situaciones que pueden afectar al genotipo antes de que se estabilice, como por ejemplo que otras criaturas lo empujen y no se pueda estabilizar. Para evitar estas situaciones, esta plataforma dispone de un conjunto de funciones y criterios que ayudan a disminuir la componente aleatoria de las simulaciones, estas funciones están implementadas tanto para la GUI como la CLI [4]. Las funciones que se han de configurar para evitar estas situaciones son:

-Simulación de la red neuronal: Esta función indica el momento en el que se pone en funcionamiento el sistema del cerebro en la simulación. Hay tres posibles opciones: que el cerebro no se active, que lo haga inmediatamente o que suceda después de la estabilización. La última opción es la que garantiza que el análisis del sistema de control del genotipo no va a estar afectado por el tiempo antes de la estabilización del genotipo evitando así medidas erróneas de su comportamiento.

-Tiempo de muestreo del comportamiento: Determina la frecuencia con la que se realiza el análisis de la actuación del individuo. Es el tiempo que transcurre entre cada evaluación.

-Tiempo de muestreo de espera: Intervalo de muestreo del genotipo antes de su estabilización. El período de tiempo transcurrido hasta que el genotipo deja de tener vibraciones iniciales.

-Tiempo máximo de espera antes de eliminar a un genotipo⁶⁷: Máximo tiempo de muestreo de espera de la simulación, es el tiempo máximo que tiene el genotipo para estabilizarse, si durante este período no lo consigue será eliminado. Se mide en función del tiempo de muestreo.

-Distancia permitida para ser estable: Tanto la distancia como la velocidad se calculan como la diferencia de longitud respecto a la posición del centro de gravedad. Para considerar que un genotipo es estable, la distancia que ha recorrido con respecto a la posición inicial de su centro de gravedad no puede ser superior a uno (por defecto es 0.01); Si esta distancia es superior se considera que el genotipo se ha desestabilizado y por tanto no es válido para la simulación y es eliminado.

⁶⁶ Este período es el tiempo de estabilización y hace referencia al intervalo en el que el genotipo no realiza ningún movimiento significativo, es decir, hasta que todas las vibraciones iniciales han desaparecido.

⁶⁷ Si el tiempo de muestreo de espera es menor que el de comportamiento, muchos genotipos serán eliminados aun pudiendo haberse estabilizado, ya que si se evalúan cada un pequeño período de tiempo y se supera el tiempo máximo de espera

-Cálculo del comportamiento: Esta función indica el momento en el que se empieza a recopilar la información sobre el comportamiento del genotipo durante la simulación. Hay tres tipos: que el cálculo no se realice, es decir, que no haya un análisis de la actuación del individuo; que se realice inmediatamente, sin esperar a que el genotipo se estabilice. El último caso, que el cálculo se produzca después de su estabilización. En esta situación se realizará el cálculo del comportamiento después de la estabilización para evitar la información errónea que pueda producirse por las vibraciones iniciales del genotipo.

Una situación a evitar es la de activar la función del comportamiento de la red neuronal en *inmediata* y el *cálculo del comportamiento después de la estabilización*; provoca que el genotipo se ponga en movimiento inmediatamente y por tanto nunca se llegue a la situación de estabilización, porque confundirá las vibraciones iniciales con los movimientos del genotipos, provocando que el cálculo de su comportamiento no llegue nunca a producirse. Para evitar todos los inconvenientes mencionados anteriormente, se han fijado los valores de los parámetros de estabilización en la Tabla 5.2-1:

Tabla 5.2-1 Relación de parámetros configurados

Parámetro	Valor Fijado
Simulación de la red neuronal	Después de la estabilización
Tiempo de muestreo de comportamiento	100
Tiempo de muestreo de espera	100
Tiempo máximo de espera antes de Eliminar a un genotipo	100
Distancia permitida para ser estable	0.01
Cálculo del comportamiento	Después de la estabilización

5.2.5 Evolución neuronal

Consiste en hallar la configuración de la red neuronal más adecuada, en el apartado 2.6.2, se explica detalladamente la red neuronal de los genotipos y su importancia e influencia en su comportamiento.

Es el conjunto de experimentos mediante los que se determina el diseño del sistema de control del genotipo que favorezca su velocidad de desplazamiento. Para ello se realizan una serie de ensayos que crean simulaciones que evolucionan el cerebro de los genotipos.

Un ejemplo esquemático de lo que se produce en la simulación evolutiva se puede observar en la siguiente figura, Figura 5.2-7, donde las “X” simbolizan las neuronas y las “Z” los músculos y las distintas conexiones y pesos que las relacionan son la caja negra en la que se engloban todas las posibles conexiones. En las pruebas 1, 2,3 se determinan la estructura de esta *caja negra*.



Figura 5.2-7 Caja negra

En la evolución las conexiones entre las neuronas y los músculos son configuradas para favorecer el objetivo de la evolución y que los sujetos permanezcan en la población.

5.2.6 Evolución morfológica

Grupo de experimentos cuya meta es encontrar la disposición de las extremidades que incrementa la velocidad del individuo. Al igual que en las pruebas anteriores, se ejecutan una serie de simulaciones evolutivas que afectan a la estructura morfológica del genotipo.

La diferencia con los ensayos anteriores, se encuentra en la configuración de la genética; el resto de parámetros se han mantenido constantes, mismo número de capacidad y algoritmos genéticos. Se ha modificado la genética de los genotipos para fijar la evolución en la modificación la longitud de las extremidades, pero manteniendo intacta la configuración neuronal.

La evolución se ha centrado en el intercambio de posiciones entre los distintos nexos. Las extremidades y nexo troncal del genotipo han pasado de una posición a otra en las distintas generaciones y se ha analizado su comportamiento durante las simulaciones. Con la intención de probar cuál era la que más favorecía al aumento de velocidad y distancia recorrida por cada uno de ellos durante su simulación.

5.2.7 Evolución completa, morfológica y neuronal.

Uno de los objetivos del experimento es analizar la influencia de distintos tipos de evoluciones en los modelos robóticos. Para ello y en último lugar, se ha realizado un experimento que incluye una evolución completa, es decir, tanto del cerebro como cuerpo del individuo simultáneamente.

Con este fin, se ha configurado la genética, permitiendo la alteración de ambos módulos del genotipo siguiendo los mismos criterios que se han utilizado para cada uno de los experimentos anteriores, intercambio de nexos y variación de pesos y conexiones entre neuronas.

Se ha realizado un estudio de como afecta la presión de selección al desarrollo y avance de la simulación. Al igual que en las pruebas anteriores se han ejecutado tres simulaciones, cada una de ellas con una capacidad distinta de número de individuos máximos en el conjunto de genotipos y número máximo de criaturas simuladas.

5.3 Resultados

En este subapartado se detalla la relación entre las pruebas realizadas, los parámetros que han influido en cada una y los resultados obtenidos. Se clasifican en función del tipo de evolución (neuronal, morfológica o completa) y los parámetros de la simulación configurados; distinguiendo entre pruebas en las que se analiza la influencia de la presión de selección y pruebas que afectan a los mecanismos de reproducción, generación de descendencia (5 a y 5b).

Se establece un criterio de validez para eliminar aquellas simulaciones que no contribuyen al objetivo del trabajo. Antes del desarrollo de cada una de las pruebas se fijan los valores mínimos

que deben de cumplir los robots evolucionados para que las pruebas sean válidas. Cada uno de estos valores depende del modelo inicial que se toma como referencia.

5.3.1 Simulaciones evolutivas neuronales

5.3.1.1 Estudio previo de resultados del experimento 1

El modelo inicial que se utiliza como base para estas pruebas ha sido bautizado como FLINK. Este modelo es fruto del estudio desarrollado en el apartado 5.2.1. En la Tabla 5.3-1 se fijan los criterios mínimos de validez de las pruebas. La velocidad y distancia recorrida por FLINK, son los valores recogidos durante su simulación.

Se han establecido como valores mínimos que tienen que cumplir los genotipos de las simulaciones para ser válidas. Estos criterios son: aumentar como mínimo la velocidad y distancia recorrida por los distintos genotipos durante la simulación un 20%.

Tabla 5.3-1 Valores mínimos

	FLINK	FLINK' 20%	FLINK' 30%
VELOCIDAD [Pasos⁶⁸/segundos]	0,00588	0,00706	0,00764
DISTANCIA [Pasos]	29,41	35,292	38,233
AJUSTE	14,71	17,65	19,12
GENERACIÓN	0	x	x

Los valores de Ajuste global se han calculado utilizando la función de evaluación que se muestra a continuación (Ecuación 5.3-1):

$$Ajuste_i = (Velocidad_i * 0,5) + (Distancia_i * 0,5)$$

Ecuación 5.3-1 Ecuación de Ajuste⁶⁹

Como el valor de todos los pesos debe ser igual a 1, se ha activado en cada una de las simulaciones una función que modifica los valores de los pesos establecidos por el usuario para que la suma de todos ellos equivalga siempre a uno y así la función de evaluación esté equilibrada.

5.3.1.2 Resultado de pruebas de evolución neuronal

En la siguiente Tabla 5.3-2, se muestra cada una de las simulaciones evolutivas realizadas y los principales parámetros que han sido configurados en cada una de ellas junto a los genotipos iniciales que se han utilizado.

⁶⁸ El cálculo de la velocidad y la distancia recorrida se hace en base al desplazamiento del centro de gravedad del genotipo, en el caso de la distancia, y al ratio entre el desplazamiento y el tiempo transcurrido durante ese desplazamiento (segundos) para la velocidad.

⁶⁹ Donde i hace referencia al genotipo evaluado.

Tabla 5.3-2 Relación de experimentos de evolución neuronal

Parámetros	EXP 1	EXP 2	EXP 3
Población inicial	1 genotipo: FLINK	1 genotipo: FLINK	1 genotipo: FLINK
Capacidad conjunto de genotipos	10	5	3
Criaturas simuladas	5	5	3
Situación inicial de los genotipos	Central	Central	Central
O G: Copia [0 100]	20	20	20
O G: Mutación[0 100]	64	64	64
O G: Cruce [0 100]	16	16	16
AG de Selección	Ajuste proporcional	Ajuste proporcional	Ajuste proporcional
AG de Reemplazo	Inversamente proporcional	Inversamente proporcional	Inversamente proporcional
Función de Evaluación	$(\text{Velocidad} * 0,5) + (\text{distancia} * 0,5)$	$(\text{Velocidad} * 0,5) + (\text{distancia} * 0,5)$	$(\text{Velocidad} * 0,5) + (\text{distancia} * 0,5)$
Normalización de la función de evaluación	si	si	Si
Estancamiento de la simulación (número de pasos)	500	500	500
Alteración genética cuerpo	0	0	0
Alteración genética cerebro	0	0	0
Alteración genética conexiones(Peso en la evolución [0 100])	Nueva conexión: 25 Borrar conexión:25 Peso :50	Nueva conexión: 25 Borrar conexión:25 Peso :50	Nueva conexión: 25 Borrar conexión:25 Peso :50
Minimo Ajuste de mejora	20%	20%	20%
Nombre mejores genotipos evolucionados	FLINK_EVOL_EXP1	FLINK_EVOL_EXP2	FLINK_EVOL_EXP3
Parámetros modificados	Capacidad & Conexiones	Capacidad & Conexiones	Capacidad & Conexiones

Estas simulaciones han generado los siguientes modelos que cumplen los criterios establecidos en la Tabla 5.3-1. Los resultados obtenidos por los genotipos evolucionados y la generación en la que lo han conseguido aparece descrito en la Tabla 5.3-3 .

Tabla 5.3-3 Resultados experimentos evolución cerebral

	FLINK_EVOL_EXP1	FLINK_EVOL_EXP2	FLINK_EVOL_EXP3
VELOCIDAD [Pasos/segundos]	7,13E-3	7,18E-3	7,45E-3
DISTANCIA [Pasos]	35,64	35,90	37,22
AJUSTE	17,82	17,95	18,62
GENERACIÓN	6	5	3
%Velocidad FLINK	21,13%	22,05%	26,54%
%Distancia FLINK	21,18%	22,09%	26,58%

Como se puede observar en la Tabla 5.3-3, el robot que alcanza antes los mínimos requeridos, es el creado en EXP 3. En la tercera generación alcanza una velocidad y una distancia recorrida un 26% superior al del genotipo inicial. Por el contrario en los EXP 1 y EXP 2 estos valores son inferiores y se producen en generaciones mayores.

La causa de esta evolución rápida y óptima es debida al aumento en la presión de selección de los individuos del experimento 3. La capacidad de la población es de un máximo de 3 individuos. Al disminuir la capacidad aumenta la presión de selección provocando que sólo se seleccionen los mejores modelos para la siguiente generación. El genotipo *IKKI*⁷⁰ pasará a ser el genotipo inicial de los siguientes experimentos (evolución morfológica, 5.3.2.1). Por ser el genotipo con la configuración neuronal que más rápido se desplaza.

Todos los modelos han sufrido modificaciones en el cerebro, en la Figura 5.3-1, se observa la evolución neuronal con respecto al modelo inicial, de los mejores robots desarrollados en cada una de las tres pruebas. En esa misma figura, el genotipo numerado con un 1, la neurona constante 10 tiene dos conexiones; sin embargo, en la siguiente evolución, el modelo 2, aumenta a tres. El mejor de los modelos, IKKI, conserva las tres, pero el número de conexiones con la neurona constante 1 disminuye a tres, en uno de los músculos giratorios y la neurona asociada a él se desconecta de la red neuronal principal.

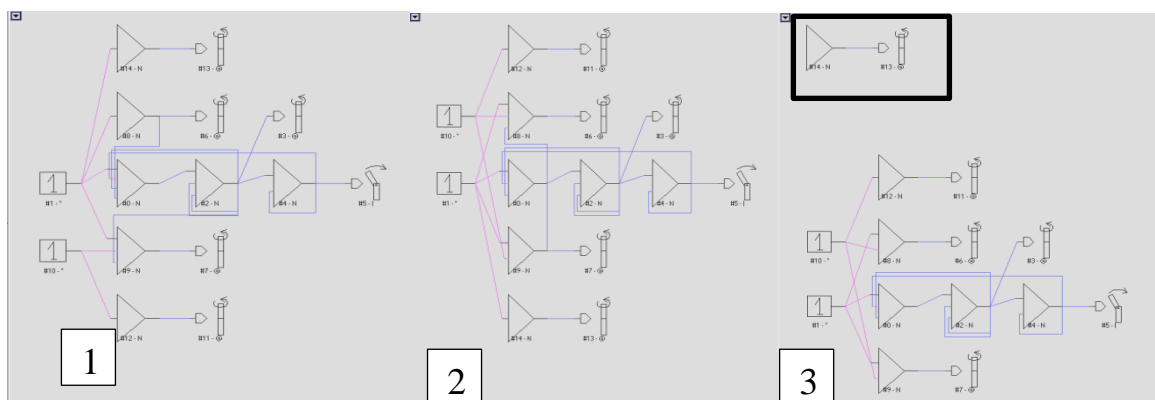


Figura 5.3-1 Cerebro de los modelos evolucionados⁷¹

Como se puede observar, uno de los músculos giratorios se ha desconectado de la red principal, al igual que en el experimento de Ingo Rechenberg⁷², el modelo más apto, no tiene porque seguir una evolución lógica. Esta es una de las ventajas de los algoritmos genéticos, muestran un conjunto de posibilidades que desde un punto de vista lógico son impensables.

⁷⁰ Se ha cambiado el nombre del genotipo *FLINK_EVOL_EXP3* por el de *IKKI* para evitar confusiones con los nombres de los experimentos

⁷¹ Capturas de pantalla obtenidas de la aplicación FRED.

⁷² Para más información consultar el punto 2.7.8.

5.3.2 Simulaciones evolutivas morfológicas

5.3.2.1 Pruebas de evolución morfológica

Las pruebas realizadas en este apartado tienen como objetivo encontrar el genotipo que se desplace con mayor velocidad evolucionando la longitud de sus extremidades. Se han lanzado distintas simulaciones atendiendo a diversos parámetros. La clasificación es la siguiente:

- Parámetros relacionados con la capacidad de la población
- Parámetros que afectan a los mecanismos de generación de descendencia

5.3.2.2 Estudio previo de resultados del experimento 2

Al igual que en el caso anterior, los valores mínimos de velocidad y distancia recorrida por el genotipo durante la simulación, tienen que ser como mínimo un 20% superior a la del genotipo origen, que en este caso es IKKI, la evolución de FLINK. En la Tabla 5.3-4 se detallan.

Tabla 5.3-4 Valores mínimos del experimento 2

	FLINK	IKKI	IKKI' 20%	IKKI' 30%
VELOCIDAD [Pasos/segundos]	5,91E-3	7,00E-3	9,01E-3	9,70E-3
DISTANCIA [Pasos]	29,41	37,23	44,67	48,35
AJUSTE GLOBAL	14,70	18,61	22,34	24,20
GENERACIÓN	0	3	x	x

Se han mantenido los mismos criterios de estabilización y función de evaluación que en el caso anterior.

5.3.2.3 Simulaciones relacionadas con la capacidad de la población

En los siguientes experimentos, se ha modificado la capacidad del conjunto de genotipos para ver como afecta la presión de selección a la evolución y resultados de la simulación. En la Tabla 5.3-5 se resume las pruebas realizadas. El objetivo es transformar el cuerpo del genotipo, variar la longitud y orientación de los nexos, con la finalidad de conseguir un desplazamiento más rápido por un entorno artificial.

Tabla 5.3-5 Pruebas de evolución morfológica, parte 1

	EXP 4 a	EXP 4 b	EXP 4 c
Población inicial	1 genotipo: IKKI	1 genotipo: IKKI	1 genotipo: IKKI
Capacidad conjunto de genotipos	10	5	3
Criaturas simuladas	5	5	3
Situación inicial de los genotipos	Central	Central	Central
O G: Copia	20	20	20
O G: Mutación	64	64	64
O G: Cruce	16	16	16
AG de Selección	Ajuste Proporcional	Ajuste Proporcional	Ajuste Proporcional
AG de Reemplazo	Inversamente proporcional	Inversamente proporcional	Inversamente proporcional
Función de Evaluación	(Velocidad*0,5)+ (Distancia*0,5)	(Velocidad*0,5)+ (Distancia*0,5)	(Velocidad*0,5)+ (Distancia*0,5)
Normalización de la función de evaluación	si	si	si
Estancamiento de la simulación (número de pasos sin mejoras)	500	500	500
Alteración genética cuerpo (Peso en la evolución [0 100])	Posición articulaciones: 100	Posición articulaciones: 100	Posición articulaciones: 100
Alteración genética cerebro	0	0	0
Alteración genética conexiones	0	0	0
Datos	0	0	0
Minimo Ajuste de mejora	20%	20%	20%
Nombre genotipos evolucionados	IKKI 4, IKKI 5, IKKI 6, IKKI 7, IKKI 8	IKKI 9, IKKI 10, IKKI 11	IKKI 12, IKKI 13
NOTAS	Se para en el EXP8 generación 30	Se para en el EXP13 generación 31	Se para en el EXP11 generación 42
Parámetros modificados	CAPACIDAD & CUERPO		

Las pruebas EXP 4 a, EXP 4 b y EXP 4 c se han ejecutado siguiendo la misma dinámica que en el experimento anterior (5.3.1.2). Variando el número de individuos por población en cada una de las simulaciones como criterio de distinción.

Los resultados siguen las mismas directrices que en el caso anterior, para considerar una simulación válida, ha de cumplir como mínimo lo detallado en la Tabla 5.3-1. Los siguientes genotipos han sido evolucionados a partir del genotipo más válido de las pruebas anteriores (5.3.1.2), *IKKI*.

En la siguiente tabla, se muestra una relación de los genotipos creados en cada una de las pruebas y su análisis.

Tabla 5.3-6 Resultados simulación 4 a

	IKKI	IKKI' 4	IKKI' 5	IKKI' 6	IKKI' 7	IKKI'8
VELOCIDAD [Pasos/segundos]	7,45E-3	9,00E-3	9,50E-3	1,11E-2	1,42E-2	2,03E-2
DISTANCIA [Pasos]	37,23	46,14	47,77	54,67	72,33	104,61
AJUSTE	18,61	23,07	23,89	27,35	36,17	52,32
GENERACIÓN	0	7	9	11	15	30
%Aumento Velocidad IKKI		24%	28,3%	47%	94,3%	181%
%Distancia IKKI		24%	28,3%	47%	94,3%	181%
Notas						Parada
CAPACIDAD & CUERPO						

Como se muestra en la Tabla 5.3-6, la simulación 4 a ha creado un genotipo cuya velocidad y distancia recorrida es un 181% superior al genotipo origen de la simulación (*IKKI*). En este modelo, la nueva disposición de los nexos y articulaciones no se asemeja a ninguna estructura morfológica conocida y no sigue ningún patrón establecido⁷³. Es un ejemplo claro de como la evolución no desemboca de ningún patrón lógico o puede que sea el resultado de la aparición de un máximo local.

Por otro lado, los modelo generados en las pruebas 4 b se detallan en la Tabla 5.3-7:

Tabla 5.3-7 Resultados simulación 4 b

	IKKI	IKKI' 9	IKKI' 10	IKKI' 11
VELOCIDAD [Pasos/segundos]	7,40E-3	8,41E-3	9,04E-3	9,79E-3
DISTANCIA [Pasos]	37,22	42,04	45,22	48,98
AJUSTE	18,61	21,02	22,61	24,45
GENERACIÓN	0	22	23	42
%Velocidad IKKI		13%	21,5%	31,6%
%Distancia IKKI		13%	21,5%	31,6%
Notas				Parada
CAPACIDAD & CUERPO				

Al contrario que en el experimento previo, al aumentar la presión de selección, el valor de Ajuste no aumenta con respecto a la simulación anterior (Tabla 5.3-6), sino que disminuye. La posible causa de esta disminución puede deberse a un máximo local que determina e influencia la simulación hacia una dirección que no maximiza globalmente la función de evaluación provocando que la velocidad y desplazamiento del mejor genotipo, sea inferior respecto de la simulación anterior.

Por último, la relación de genotipos evolucionados y sus resultados se encuentra en la siguiente Tabla, Tabla 5.3-8.

⁷³ En [11] se encuentra un vídeo con la simulación 3D del comportamiento de IKKI'8.

Tabla 5.3-8 Resultados simulación 4 c

	IKKI	IKKI' 12	IKKI' 13
VELOCIDAD [Pasos/segundos]	7,45E-3	1,08E-2	1,34E-2
DISTANCIA [Pasos]	37,23	53,90	67,12
AJUSTE	18,62	26,96	33,57
GENERACIÓN	0	6	31
%Velocidad IKKI		44,8%	80,3%
%Distancia IKKI		44,8%	80,3%
Notas			Parada
CAPACIDAD & CUERPO			

En estas pruebas (4 c), las mejoras en los genotipos han sido más rápidas que en el caso anterior (4 b), produciéndose en la 6 generación pero aun así el valor de Ajuste sigue siendo inferior a la simulación 4 a.

En la simulación (4 c) se ha desarrollado un genotipo (Generación 6, *IKKI'12*) cuyo Ajuste era mayor que el del resto del conjunto. Se ha elegido como progenitor para la creación de descendencia, los parámetros de este genotipo que lo hacían más rápido han resultado no ser los idóneos para el aumento de la velocidad y la evolución se ha detenido. Por tanto al tomar como referencia dicho genotipo la evolución ha seguido un camino que ha derivado en el estancamiento de la función y en el alcance de un valor de Ajuste menor del que se podría haber alcanzado; confirmando la teoría del máximo local⁷⁴.

En la Figura 5.3-2, se observa una gráfica de como se ha ido desarrollando la evolución en función del valor de Ajuste⁷⁵ obtenido. La pendiente de las gráficas es inversamente proporcional a la presión de selección, es decir, a menor presión de selección mayor pendiente. Al contrario que en el experimento anterior, 5.3.1.2, donde a mayor presión de selección, el Ajuste iba en aumento y se producía en menos generaciones.

⁷⁴ Un máximo local se produce cuando se llega a una solución óptima parcial que deriva en un estancamiento de la evolución. Debido a que dicha solución no contiene las características necesarias para ser una solución global del conflicto.

⁷⁵ El valor de Ajuste se ha calculado utilizando la Ecuación 5.3-1 .

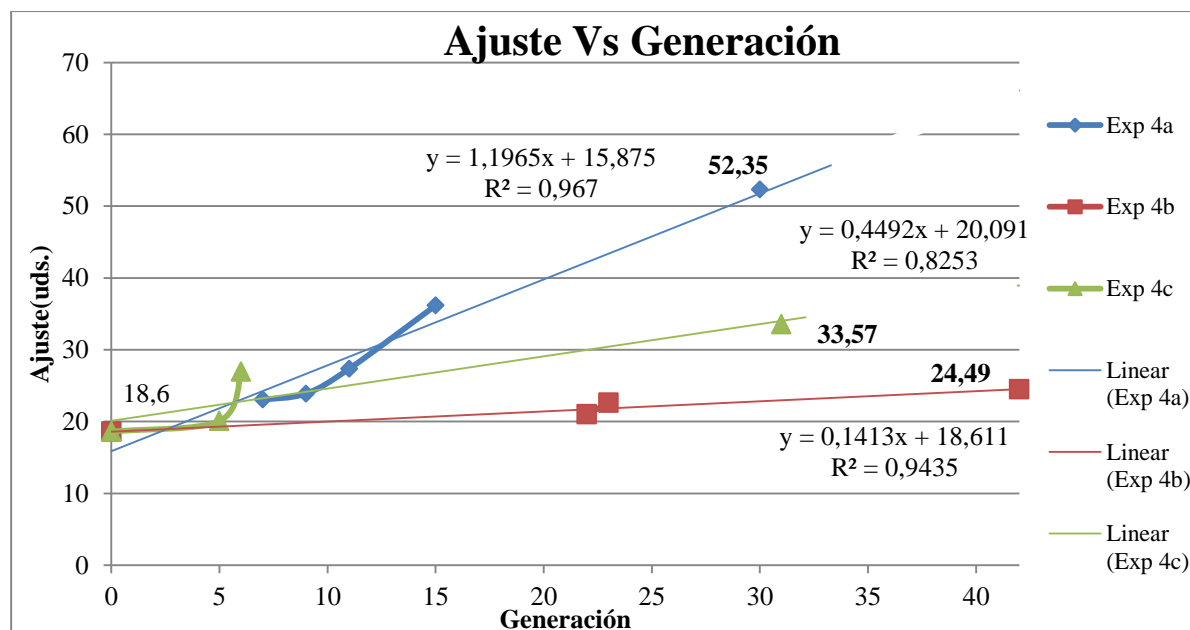


Figura 5.3-2 Resultados experimento 4 a, 4 b y 4 c

El mejor genotipo de las pruebas 4 a, 4 b y 4 c corresponde al modelo de la simulación 4 a, *IKKI'8*. Este robot ha alcanzado los valores más altos de Ajuste (ha incrementado un 181% respecto al modelo inicial) en la generación 30, como se observa en la Figura 5.3-2, Exp 4 a.

Al simular su comportamiento en Artificial Life⁶², vemos que el movimiento de desplazamiento que realiza no es el deseado, se ha degenerado⁷⁶. Prueba de que el resultado de la evolución no siempre genera un modelo plausible y lógico.

5.3.2.4 Simulaciones que afectan a los mecanismos de generación de descendencia

En esta pareja de simulaciones, se ha realizado el análisis de la influencia de los operadores genéticos en la evolución de los genotipos. Modificando los pesos de los distintos algoritmos genéticos para que la creación de descendencia tuviese mayor influencia la mutación de los padres (5 a) ó el cruce de los progenitores (5 b).

En la siguiente tabla, Tabla 5.3-9, se hace un resumen de los principales parámetros configurados en las pruebas.

⁷⁶ Para ver las simulaciones de *IKKI'8*, consultar [11]. El esquema de su estructura física se encuentra representado en la Figura 5.3-3.

Tabla 5.3-9 Pruebas de influencia de OGs, Experimentos 5 a y 5 b.

	EXP 5 a	EXP 5b
Población inicial	2 genotipos: IKKI y IKKI(Copia)	2 genotipos: IKKI y IKKI(Copia)
Genepool Capacity	10	10
Criaturas simuladas	5	5
Situación inicial de los genotipos	Central	Central
O G: Copia	0	0
O G: Mutación	80	20
O G: Cruce	20	80
AG de Selección	Ajuste Proporcional	Ajuste Proporcional
AG de Reemplazo	Inversamente proporcional	Inversamente proporcional
Función de Evaluación	(Velocidad*0,5)+(Distancia*0,5)	(Velocidad*0,5)+(Distancia*0,5)
Normalización de la función de evaluación	si	si
Estancamiento de la simulación (número de pasos sin mejoras)	500	500
Alteración genética cuerpo (Peso en la evolución [0 100])	Posición articulaciones: 100	Posición articulaciones: 100
Alteración genética cerebro	0	0
Alteración genética conexiones	0	0
Minimo Ajuste de mejora	25%	25%
Nombre genotipos evolucionados	14,15	16
Parámetros modificados	OPERADORES GENÉTICOS & CUERPO	

Del par de pruebas desarrolladas, en la Tabla 5.3-10, se encuentran los siguientes modelos evolucionados y los resultados obtenidos. En ambas pruebas se ha producido la parada de la simulación debido al estancamiento.

Tabla 5.3-10 Modelos evolucionados en las pruebas 5 a y 5 b

	IKKI	IKKI' 14	IKKI' 15	IKKI' 16
VELOCIDAD [Pasos/segundos]	7,45E-3	1,06E-2	1,38E-2	7,70E-3
DISTANCIA [Pasos]	37,23	52,84	69,25	38,51
AJUSTE	18,62	26,43	34,63	19,29
GENERACIÓN	0	6	22	4
%Velocidad IKKI		42%	86%	3,44%
%Distancia IKKI		42%	86%	3,44%
Notas			Parada	Parada
CAPACIDAD & CUERPO				

El propósito de estas pruebas ha sido el estudio de la influencia de los operadores genéticos en el proceso evolutivo. Se han modificado los pesos de los operadores genéticos de mutación y cruce para analizar como afecta el mecanismo de generación de descendencia en la evolución. En

las pruebas se han introducido dos modelos⁷⁷ iniciales y han evolucionado atendiendo a los parámetros de la Tabla 5.3-9 cuyo resultado se especifica en la Tabla 5.3-10.

En la primera simulación, 5 a, se prioriza la mutación de los padres sobre el cruce de sus genes; los resultados indican que en la sexta generación se mejora un 42% el valor de Ajuste con respecto de los padres (genotipos iniciales: IKKI y Copia de IKKI) y la simulación se interrumpe en la generación 22 debido al estancamiento⁷⁸.

Por otro lado, en la segunda simulación 5 b, la velocidad y distancia han aumentado un 3,3% con respecto a los genotipos iniciales y se ha detenido la simulación en la generación 4. El cruce de los progenitores no ha generado descendencia válida para continuar la evolución y la simulación se ha detenido.

La mutación como mecanismo de generación de descendencia en estas simulaciones ha sido más óptima que el cruce de progenitores. Como los genotipos de la población inicial poseían las mismas características, el cruce no ha aportado mejoras en la evolución provocando que los valores de Ajuste no amenten con respecto a sus progenitores desencadenando el estancamiento⁷⁸ y parada de la simulación.

5.3.2.5 Resultados finales del experimento de evolución morfológica

Realizadas todas las pruebas se observa que el mejor genotipo de todas las de evolución morfológica ha sido IKKI'8, ha aumentado un 181% su valor de Ajuste respecto de FLINK, el genotipo inicial. A continuación se muestra un esquema de las vistas de la nueva estructura del modelo evolucionado (Figura 5.3-3). La estructura no posee un orden lógico y no se asemeja a ningún modelo existente en la naturaleza. Una de las extremidades⁷⁹ ha sufrido un cambio de orientación y posición que ha modificado el movimiento del modelo.

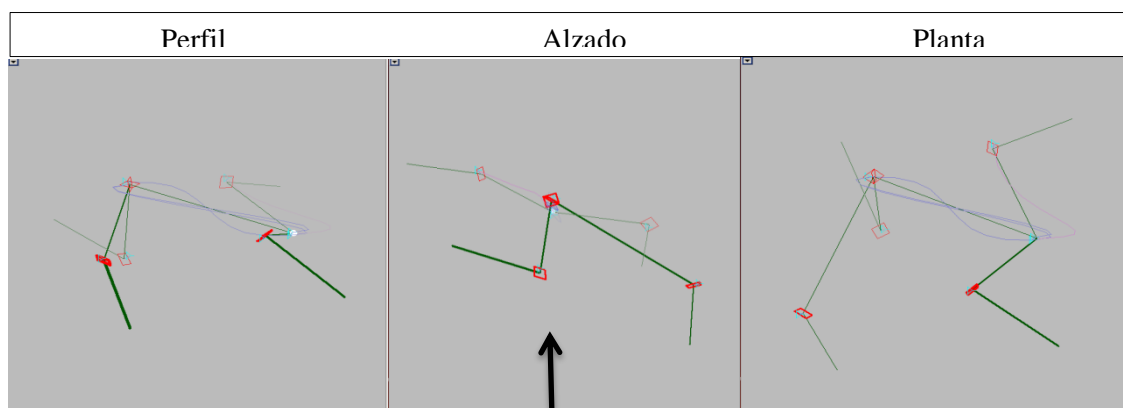


Figura 5.3-3 Esquema de la estructura de IKKI'8 obtenida del programa Framsticks GUI.

En un principio el objetivo de este modelo era imitar el comportamiento animal pero en esta nueva estructura, a pesar de ser la más rápida, se ha producido un cambio en su

⁷⁷ A diferencia del resto de experimentos, se han introducido dos genotipos para que se produzca el cruce de los progenitores.

⁷⁸ El estancamiento se produce cuando durante un período de tiempo el valor de ajuste de los individuos no mejora con respecto al de sus progenitores y el resto de individuos de la población.

⁷⁹ Dicha extremidad aparece marcada con una flecha en la Figura 5.3-3.

comportamiento⁷⁶, degenerando su movimiento. Como no cumple los objetivos del trabajo, IKKI'8 ha sido descartado como solución final.

El siguiente modelo evolucionado que sí cumple con los objetivos es IKKI'15⁸⁰; realiza un desplazamiento natural y ha aumentado un 86% su velocidad y distancia recorrida con respecto al prototipo inicial. Este genotipo se ha desarrollado en la simulación 5 a y pertenece a la vigésimo segunda generación de IKKI.

En la Figura 5.4-1 están representadas las vistas de este modelo y su configuración neuronal, al producirse la evolución sólo de su estructura física, su sistema de control se mantiene constante y es igual que el de IKKI, el modelo inicial, que aparece representado en la Figura 5.3-1, modelo 3.

5.3.3 Simulaciones evolutivas completas

5.3.3.1 Estudio previo de resultados de la evolución completa

En estas simulaciones se producirá una evolución neuronal y estructural de los modelos, los parámetros que se han configurado en cada simulación se detallan en la Tabla 5.3-11. Antes de ello, se realiza un estudio previo siguiendo las siguientes premisas:

- Cuanto mayor es la presión de selección, los genotipos evolucionados alcanzan mejor Ajuste que los genotipos de otras pruebas con menor presión de selección.
- El número de generaciones que tardan en alcanzar estos objetivos disminuye también.

Es decir a mayor presión de selección, la generación cuyos genotipos superan los valores mínimos debería ser menor. Al disminuir el número de criaturas simuladas, el avance de la simulación disminuye. Para analizar el comportamiento de las criaturas hay que esperar más tiempo ya que la capacidad de la población temporal es menor y son simuladas en grupos reducidos; aunque esto no afecta directamente a la presión de selección.

En la Figura 5.3-4, se representa el estudio previo a modo de gráfica, de como debería de ser el valor de Ajuste de los individuos a medida que aumentan las generaciones. Cada una de las series de la gráfica, están ordenadas de mayor a menor presión de selección, siendo la Serie Ajuste 1 la prueba con mayor presión de selección. Y el valor de cada serie es el porcentaje de incremento con respecto del valor de Ajuste de FLINK (Ecuación 5.3-2).

Se puede observar que el porcentaje de incremento del valor de Ajuste es directamente proporcional a la presión de selección. En las pruebas de evolución completa detalladas a continuación, se comparan los resultados obtenidos en los experimentos con el estudio previo y se genera una gráfica, Figura 5.3-7, con los valores reales obtenidos de incremento de Ajuste de los genotipos a medida que avanzan las generaciones.

⁸⁰La configuración física y neuronal de este genotipo aparecen codificados en formato f0 en el Anexo junto con los planos de su estructura física. Esta disponible una simulación de su comportamiento en [11].

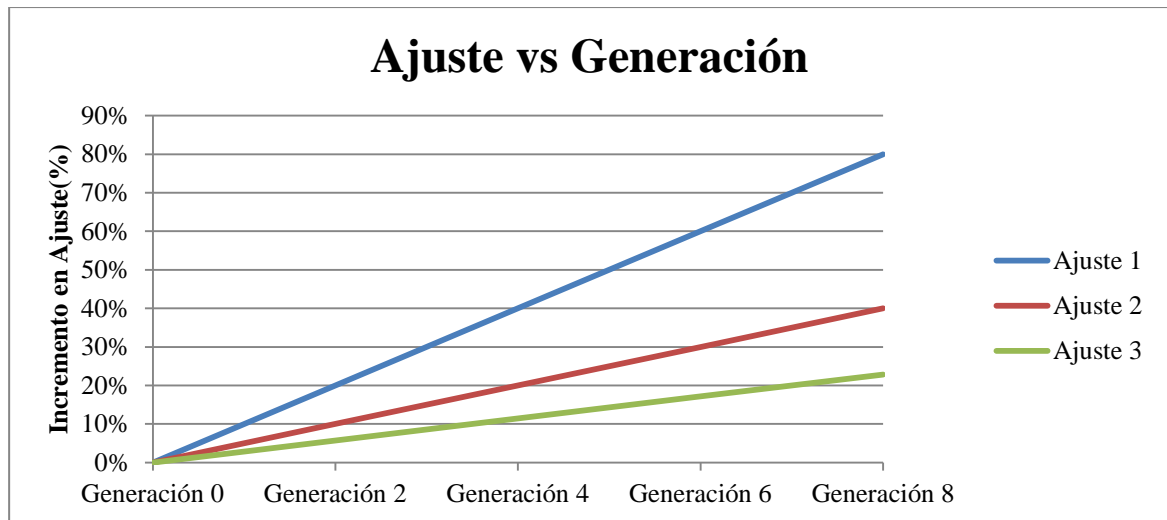


Figura 5.3-4 Gráfica de Ajuste vs Generación

$$Incremento(\%) = \frac{(Ajuste\ x - Ajuste\ FLINK)}{Ajuste\ x} * 100$$

Ecuación 5.3-2 Ecuación de incremento de Ajuste

Para realizar este estudio se ha utilizado el modelo inicial del primer experimento, FLINK y se han mantenido los criterios de valores mínimos utilizados en el resto de las pruebas (Tabla 5.3-1).

En el siguiente apartado se encuentra una tabla (Tabla 5.3-11) con todas las premisas que han sido configuradas en cada una de las distintas pruebas del experimento y el nombre de los genotipos que han cumplido los valores mínimos establecidos en la Tabla 5.3-1⁸¹.

5.3.3.2 Resultado de pruebas de evolución completa

En estas últimas pruebas se ha modificado la genética para que se puedan alterar tanto las conexiones como el cuerpo del individuo simultáneamente. Se han asignado pesos a cada uno de los parámetros que se quieren evolucionar. La genética asigna un peso a cada uno de los parámetros que pueden alterarse durante la evolución, la suma de todos ellos tiene que ser igual a 100 porque si no se asigna aleatoriamente la diferencia al resto de parámetros automáticamente. Los parámetros que se configurado se resumen en la Tabla 5.3-11.

⁸¹ Al utilizar el mismo modelo inicial que en las pruebas e evolución neuronal, los valores mínimos se mantienen.

Tabla 5.3-11 Resumen de parámetros de las pruebas de evolución completa

	EXP' 1	EXP' 2	EXP' 3
Población inicial	1 genotipo: FLINK	1 genotipo: FLINK	1 genotipo: FLINK
Capacidad conjunto genotipos	10	5	3
Criaturas simuladas	5	5	3
Situación inicial	Central	Central	Central
O G: Copia	20	20	20
O G: Mutación	64	64	64
O G: Cruce	16	16	16
AG de Selección	Ajuste proporcional	Ajuste proporcional	Ajuste proporcional
AG de Reemplazo	Inversamente proporcional	Inversamente proporcional	Inversamente proporcional
Función de Evaluación	(Velocidad*0,5)+ (Distancia*0,5)	(Velocidad*0,5)+ (Distancia*0,5)	(Velocidad*0,5)+ (Distancia*0,5)
Normalización de la función de evaluación	si	si	si
Estancamiento de la simulación (número de pasos sin mejoras)	500	500	500
Alteración genética cuerpo	Posición articulacione: 50	Posición articulaciones: 50	Posición articulaciones: 50
Alteración genética cerebro	0	0	0
Alteración genética conexiones	Nueva conexión: 16,6 Eliminar conexión:16,6 Peso conexión:16,6	Nueva conexión: 16,6 Eliminar conexión:16,6 Peso conexión:16,6	Nueva conexión: 16,6 Eliminar conexión:16,6 Peso conexión:16,6
Minimo Ajuste de mejora	20%	20%	20%
Parámetros modificados	Capacidad & Conexiones& Cuerpo		

Las características del comportamiento de los genotipos evolucionados en cada una simulación del experimento, se describen a continuación. Se ha bautizado como *IGOR* a todos los genotipos que han sido creados durante este experimento.

En la Tabla 5.3-12 se encuentran los modelos evolucionados en la prueba EXP'1. En esta prueba se han mantenido los mismos parámetros que en el ensayo EXP 4 a y EXP 1, excepto los pesos de la genética, siendo la prueba con la presión de selección más baja.

Tabla 5.3-12 Resultado Simulación EXP'1

	FLINK	IGOR'1	IGOR'2	IGOR'3	IGOR'4	IGOR'5
VELOCIDAD [Pasos/segundos]	5,88E-3	7,31E-3	7,01E-3	5,02E-3	6,01E-3	9,53E-3
DISTANCIA [Pasos]	29,41	36,78	38,19	24,97	29,70	47,81
AJUSTE	14,70	18,4	19,1	12,5	14,8	23,92
GENERACIÓN	0	2	4	6	8	33
%Velocidad FLINK		25%	29,8%	-15,2%	0,96%	62,5%
%Distancia FLINK		25,1%	30%	-15,2%	1%	62,6%

En la siguiente tabla, Tabla 5.3-13, se encuentran los robots generados durante la prueba EXP'2, cuyas características son idénticas a las pruebas EXP 4 b y EXP 2 a excepción de la genética, que determina el tipo de evolución.

Tabla 5.3-13 Resultado Simulación EXP'2

	FLINK	IGOR''1	IGOR''2	IGOR''3	IGOR''4	IGOR''5
VELOCIDAD [Pasos/segundos]	5,88E-3	6,01E-3	7,10E-3	6,50E-3	6,42E-3	1,07E-2
DISTANCIA [Pasos]	29,41	29,37	35,98	32,67	32,12	53,09
AJUSTE	14,70	14,7	18,0	16,3	16,1	26,5
GENERACIÓN	0	2	4	6	8	26
%Velocidad FLINK		-0,16%	20,3%	11,05%	11,05%	81,4%
%Distancia FLINK		-0,13%	22,3%	11,1%	11,1%	80,5%

Por último, la Tabla 5.3-14, resume los genotipos que han sido generados en la simulación de la prueba EXP'3. Al igual que en las pruebas anteriores (EXP'1 y EXP'2) los parámetros de estas simulaciones se han mantenido iguales que en las pruebas EXP 3 y EXP 4 c a diferencia de la genética. Esta prueba posee la presión de selección de los genotipos más alta. Teniendo en cuenta el estudio previo realizado en el apartado anterior, debería generar los modelos con mayor valor de Ajuste y más óptimos de todo el conjunto de pruebas de evolución morfológica.

Tabla 5.3-14 Resultado de la simulación de la prueba EXP'3

	FLINK	IGOR'''1
VELOCIDAD [Steps/second]	5,88E-3	5,87E-3
DISTANCIA [Steps]	29,41	29,44
AJUSTE	14,70	14,723
GENERACIÓN	0	1
%Velocidad FLINK		-0,064%
%Distancia FLINK		0,10%
Notas		Parada

Analizados todos los modelos robóticos, se determina que el mejor genotipo ha sido generado en la simulación EXP'2. Este modelo ha aumentado un 80% su valor de Ajuste con respecto al genotipo inicial, FLINK. Ha sido bautizado como IGOR''582 y su actuación está resumida en la Tabla 5.3-13.

Esta mejora en la habilidad de desplazamiento y velocidad se ha debido a una evolución en la red neuronal y en la estructura del genotipo. En la Figura 5.3-5, se observan los cambios que se han producido en la estructura del modelo debidos a la evolución; si se compara con la Figura 5.2-1, los nexos que conectan las extremidades con las articulaciones 0 y 3 han sufrido cambios tanto en la longitud, produciéndose una disminución, como en su orientación respecto de las articulaciones a las que se unen. Los nexos que forman las patas de los genotipos se han hecho más largos y el ángulo que los une a las articulaciones también ha sido modificado. En el alzado de la Figura 5.3-5, la articulación enumerada con un "1", presenta la evolución que sucedería en la naturaleza, la elevación de la extremidad y el aumento de longitud de las patas.

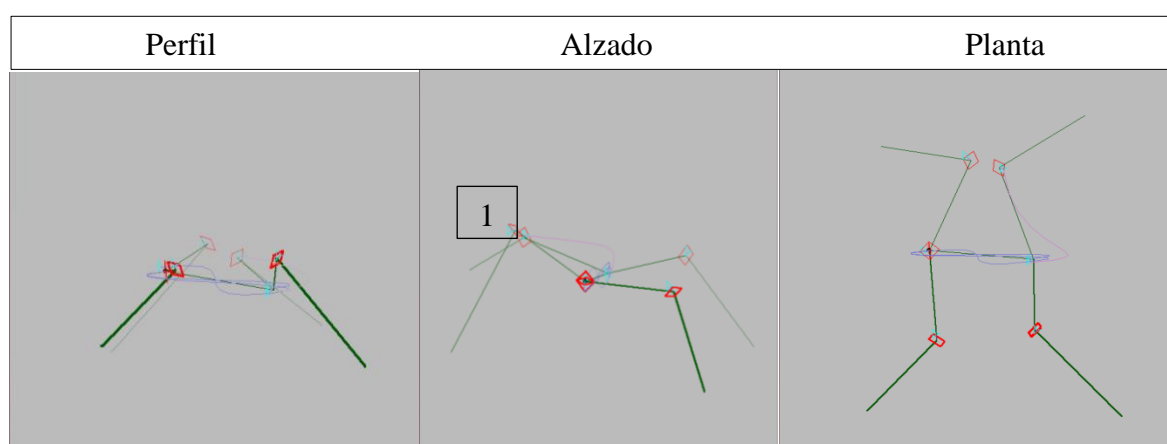


Figura 5.3-5 Vistas de la estructura del genotipo IGOR''5

En cuanto al cerebro, como en la genética se había establecido que sólo se pudieran realizar cambios en las conexiones y pesos de la red neuronal, el número y tipo de neuronas se ha mantenido constante pero han variado los pesos que se le asigna a cada una de las neuronas y han aumentado el número de conexiones. En la Figura 5.3-6, se marca una de las conexiones evolucionadas, se ha añadido una conexión de realimentación en una de las neuronas.

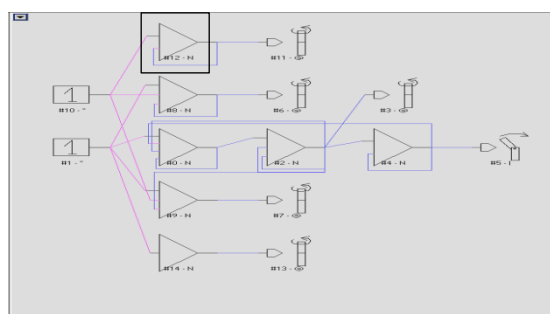


Figura 5.3-6 Cerebro de IGOR''5

⁸² Tanto el código de IGOR''5 como su plano esquemático se encuentra detallado en el Anexo.

5.3.3.3 Análisis del Ajuste en relación a la generación de genotipos

Teniendo en cuenta el estudio previo del subapartado 5.3.3.1, se ha realizado un análisis de la relación entre el valor de Ajuste de los genotipos y el número de generaciones. Para poder comparar el comportamiento que se predijo con los resultados de los ensayos. Con esta intención se han recogido los datos obtenidos en las generaciones dos, cuatro, seis y ocho de cada una de las pruebas para analizar como han evolucionado los modelos en cada una de ellas.

En la Figura 5.3-7, se muestra una gráfica con el comportamiento de cada una de las simulaciones. Se representa el valor de Ajuste individual de cada uno de los genotipos a lo largo de las generaciones de cada una de las pruebas. Se han incluido líneas de tendencia polinómicas de cuarto grado para observar la tendencia de la evolución.

Las fórmulas utilizadas para calcular el valor de Ajuste se explican en la Ecuación 5.3-1. Al poseer los valores reales, no se han predicho los valores de las siguientes generaciones como en la Figura 5.3-4, utilizada en el estudio previo y donde se ha empleado la Ecuación 5.3-2 para predecir el incremento en el valor de Ajuste. Al contrario, se ha representado el valor de Ajuste del mejor de los genotipos pertenecientes a cada una de las generaciones que han sido representadas (2,4, 6 y 8) y mediante líneas de tendencia se ha observado la progresión de cada prueba.

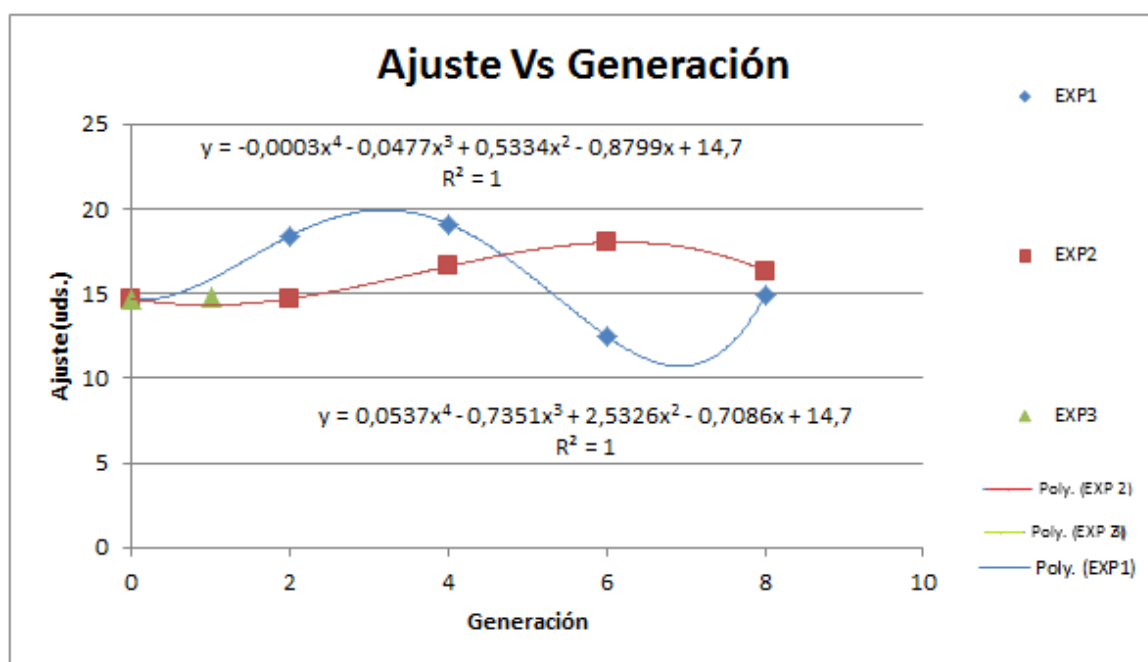


Figura 5.3-7 Análisis Evolución completa

En el ensayo EXP'1, ha habido una evolución rápida durante las primeras generaciones, luego se ha producido una disminución del Ajuste y a continuación se ha iniciado una recuperación. Si se hubiera continuado la simulación, el valor de Ajuste incrementaría, ya que se observa que la tendencia de la evolución va en aumento. Se ha continuado la simulación hasta que se ha parado en la generación 33, debido al estancamiento de la simulación. Su mejor genotipo ha sido IGOR'5, con un valor de Ajuste mayor que el del resto de genotipos creados y se ha generado en

esta última por lo que queda comprobado que la tendencia de la evolución era positiva, hacia valores de Ajuste superiores.

Sin embargo en la segunda simulación, EXP'2, la evolución sigue la misma tendencia pero es más lenta y tarda más en aumentar el valor de Ajuste. Siguiendo la línea de tendencia y comparando los resultados obtenidos, el valor de Ajuste disminuiría de nuevo y luego aumentaría hasta el valor del último genotipo antes del estancamiento. Al igual que en la prueba anterior, el mejor genotipo de toda la simulación (IGOR''5) se ha creado en las sucesivas generaciones y su valor de Ajuste es superior al de los representados.

La tercera simulación se ha detenido en la primera generación, la presión de selección era mayor que en los casos anteriores por lo que la variedad de individuos estaba más limitada, impidiendo que en pocas generaciones hubiera un incremento en el valor de Ajuste lo que ha provocado el estancamiento y el fin de la simulación.

Comparando los resultados obtenidos con el estudio previo, no se ha producido el comportamiento que se predijo. El mejor modelo no ha sido generado en las pruebas con mayor presión de selección, ha sucedido al contrario, la prueba con mayor presión de selección ha sido la que más rápido se a estancado sus los modelos desarrollados tienen los valores de Ajuste mas bajos de todo el ensayo de evolución completa.

A menor población con más variables que modificar (más regiones del espacio para explorar) la evolución se ha concentrado en aquellas variables que en un principio se creía que favorecían la evolución (mínimos locales) provocando el estancamiento rápido de la simulación. Por ejemplo, si uno de los primeros genotipos tiene un sensor de agua en una de sus articulaciones y obtiene un mayor valor de Ajuste que el resto de genotipos del conjunto, siendo la presión de selección alta, la simulación lo seleccionará para crear descendencia. Si el cromosoma del sensor de agua pasase a la siguiente generación, esta neurona no aportaría mejoras a la evolución y la simulación se acabará estancando.

Al aumentar la presión de selección en las simulaciones que pueden modificar más módulos del genotipo (cerebro y cuerpo a la vez), se ha observado que el valor de Ajuste es inversamente proporcional a la presión de selección.

5.4 Conclusiones de los experimentos

Según las pruebas realizadas, la evolución ha alcanzado mayores valores de Ajuste al realizarla por partes (evolución neuronal y evolución morfológica) que completa. Sin embargo, el comportamiento de los genotipos generados en la evolución completa se parecen más a los movimientos que podrían darse en la naturaleza que los desarrollados en la evolución por partes.

En las pruebas EXP 1, EXP 2, EXP 3, EXP 4 a, EXP 4 b, EXP 4c, EXP 5 a y EXP 5b los individuos han obtenido mejores valores de Ajuste en menos generaciones pero en varias de estas pruebas los modelos generados representan máximos locales y los individuos creados no realizan movimientos naturales.

Por el contrario en las pruebas de evolución completa obtienen valores inferiores pero los movimientos locomotores emulan a los comportamientos que se pueden dar en la naturaleza y la tendencia de su estructura física es más parecida a la evolución natural, el centro de gravedad se acerca más al suelo para ganar estabilidad, las extremidades se alargan etc.

En el estudio previo de como afecta la presión de selección al valor de Ajuste en cada generación de individuos, se predice que al aumentar la presión, el valor de Ajuste de los genotipos aumentará más rápidamente; sin embargo en las pruebas realizadas no ha ocurrido así, los mejores genotipos han aparecido en aquellas pruebas cuya presión de selección era menor que en el resto del experimento.

La posible causa de esta contrariedad se encuentra en la aparición de máximos locales y en el reducido número de individuos del conjunto de genotipos de alguna de las pruebas realizadas, al ser tan pequeña la capacidad, la diversidad disminuye provocando el estancamiento de la evolución.

En cuanto a la influencia de los operadores genéticos en la evolución, se ha demostrado que el mejor mecanismo para la generación de descendencia en los experimentos realizados es la mutación. El mejor genotipo de todas las pruebas realizadas ha sido sometido en mayor porcentaje a la mutación que en el resto de las pruebas.

Por tanto para lanzar una simulación cuyos objetivos estén alienados con lo investigado en este trabajo, los parámetros que debemos tener en cuenta son: aumentar la presión de selección y dar una mayor importancia a la mutación como operador genético encargado de la generación de descendencia; y realizar la evolución conjunta, cuerpo y cerebro del modelo evolucionado en un mismo proceso.

De todos los experimentos realizados, el mejor resultado que se ha obtenido en las pruebas ha sido IKKI'8. Es el mejor genotipo de todo el experimento, ha alcanzado la mayor velocidad y distancia recorrida de todos los creados, obteniendo el mayor valor de Ajuste. Sin embargo, el objetivo de estos experimentos es imitar el comportamiento animal a través de modelos que ejecuten movimientos plausibles. Es cierto que IKKI'8 es el genotipo más rápido, pero sus movimientos locomotores no son válidos.

Entonces el genotipo que se acerca más a la solución del conflicto es IKKI'15 con un 86% de incremento en el valor de Ajuste con respecto de FLINK; y un valor de Ajuste absoluto de 34,63. Cuyo movimiento no se ha degenerado con respecto al modelo inicial; IKKI'15 se ha convertido en el mejor genotipo de todo el experimento. Ha sido generado en las pruebas evolutivas que tenían en la mutación la prioridad para generar descendencia.

En la siguiente tabla, Tabla 5.4-1, se hace un resumen de los mejores genotipos y los resultados obtenidos.

Tabla 5.4-1 Resultado mejores genotipos

	FLINK	IKKI	IKKI'8	IKKI'15	IGOR'5
VELOCIDAD [Pasos/segundos]	5,88E-3	7,44E-3	2,03E-2	1,38E-2	1,07E-2
DISTANCIA [Pasos]	29,41	37,23	104,61	69,25	53,09
AJUSTE (valor absoluto)	14,70	18,61	52,32	34,63	26,55
Nombre simulación	EXP0	EXP3	EXP 4 a	EXP 5 a	EXP'2

IKKI'15 comparte las mismas características neuronales que IKKI, ya que en la simulación EXP 5 a sólo ha evolucionado la estructura física. En la Figura 5.4-1, se muestra la vista alzada, planta y perfil de su estructura física, donde se han producido los cambios.

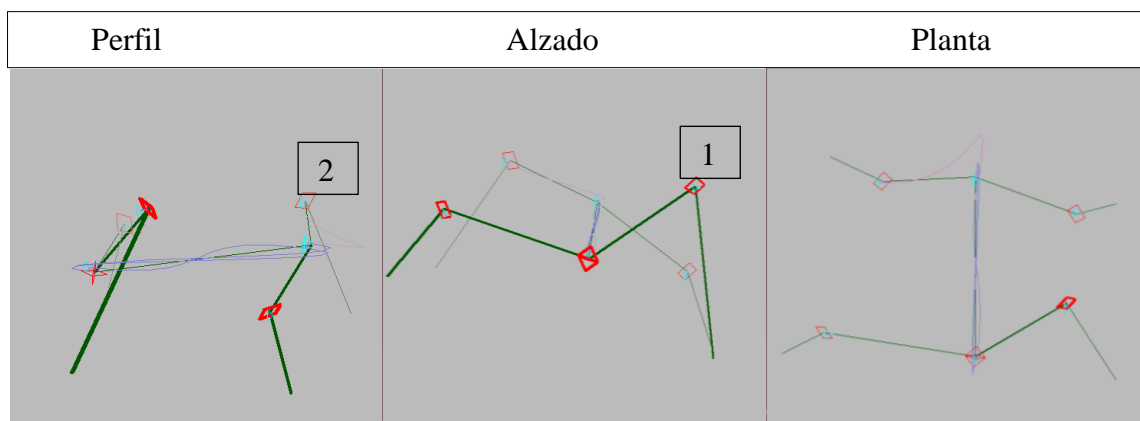


Figura 5.4-1 Vistas de IKKI'15

Las articulaciones de IKKI'1583 se han elevado para facilitar el avance del individuo, aumentando la longitud de los nexos de unión con el tronco de modelo. La articulación 1 y 2, Figura 5.4-1, representan la evolución que tendría más posibilidades de suceder en la naturaleza. En el modelo robótico IGOR''5, también se observa el aumento de longitud de una de sus extremidades (Figura 5.3-5). La elevación de las articulaciones y el descenso del centro de gravedad para facilitar el rápido desplazamiento ha sido la principal alteración física que se ha producido en los modelos.

En la Tabla Anexo-, Tabla Anexo-1 y Tabla Anexo-3 se encuentran la longitud de cada uno de los nexos de los modelos FLINK, IKKI'15 y IGOR''5 con las correspondientes variaciones en las medidas que se han mencionado. Todos los nexos del prototipo inicial tenían una longitud de una unidad, sin embargo ninguna de las extremidades de IKKI'15 y IGOR''5 tiene esa longitud.

Los modelos evolucionados han sufrido alteraciones en la posición y longitud de sus nexos y articulaciones para facilitar el desplazamiento, por tanto se han cumplido los objetivos del experimento evolutivo. En la Figura 5.4-2 y Figura 5.4-3 se ha creado un modelo robótico de la posible evolución que tendría el genotipo en la naturaleza, cómo se puede observar, el centro de gravedad se encuentra más cercano al suelo y las extremidades se han elevado.

⁸³ Tanto el plano esquemático de su estructura física como su código y medidas se encuentran en el Anexo.

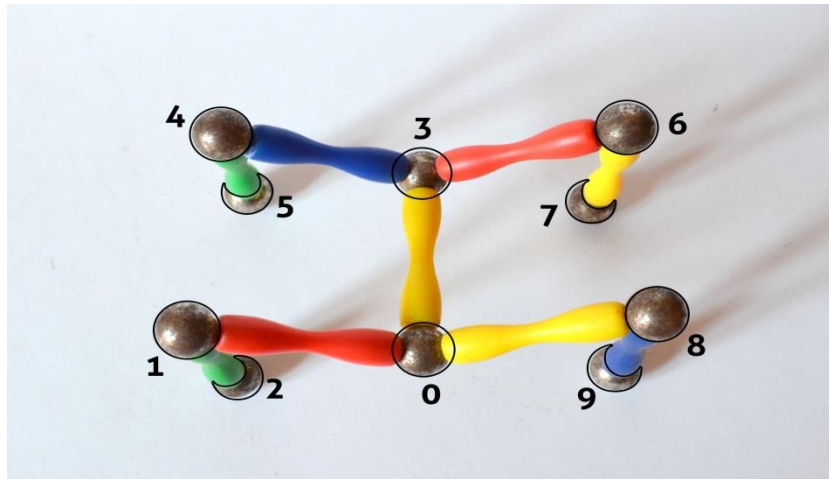


Figura 5.4-2 Planta modelo evolución natural

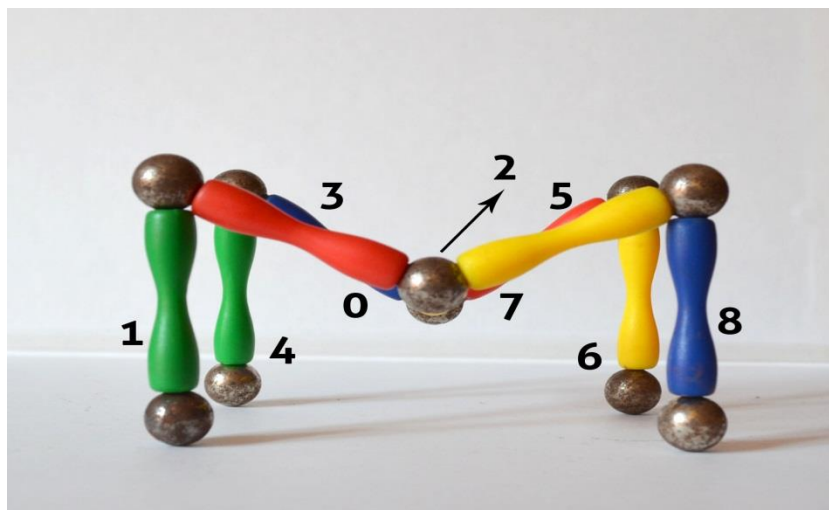


Figura 5.4-3 Alzado modelo evolución natural

5.5 Artificial Life

5.5.1 Introducción

La intención de este trabajo es encontrar el modelo robótico evolucionado que avance más rápidamente, realizando un desplazamiento que imite al que se podría dar en la naturaleza. Mediante la GUI, este análisis se queda escaso y no es preciso, por lo que se utiliza la aplicación Artificial Life, AL, de la herramienta Framsticks encargada de la simulación gráfica en tres dimensiones.

Gracias a esta herramienta se puede estudiar el comportamiento de los modelos creados y es utilizada como método de selección del mejor genotipo de todas las pruebas realizadas. Porque al poder simular su comportamiento se pueden descartar aquellos modelos cuyo movimiento se ha degenerado. A continuación se hace una descripción de sus partes y funciones principales.

5.5.2 Definición y principal uso

Artificial Life (AL), es la nueva aplicación incorporada a la plataforma Framsticks, engloba funciones tanto de diseño de genotipos como de simulación y evaluación. Posee varios módulos,

divididos en dos funciones principales, edición y simulación de genotipos. En la Figura 5.5-1 aparece el texto de bienvenida del programa.



Figura 5.5-1 Pantalla de Inicio Artificial Life

El módulo de edición no ha sido utilizado en este trabajo. Sin embargo el módulo encargado de la simulación en 3D del comportamiento de los genotipos se ha empleado para analizar y estudiar los modelos creados y los generados por la plataforma⁸⁴. La Figura 5.5-2 es una captura de pantalla de una simulación lanzada en AL.



Figura 5.5-2 Simulación en 3D en Artificial Life

Gracias a esta aplicación se ha podido estudiar el comportamiento de los modelos iniciales y genotipos creados a través de las distintas pruebas. Ha ejercido un papel determinante a la hora de elegir tanto el modelo del experimento como el mejor resultado de todas las pruebas realizadas. AL ha sido empleada como método de evaluación de la locomoción de los modelos generados en el trabajo.

Su funcionamiento es básico, hay que guardar el fichero con las propiedades codificadas del genotipo o genotipos que se quieran simular en un directorio determinado con un nombre concreto (*mobycurrent.gen*). Al iniciar el programa se carga la configuración del modelo guardado. Se introduce en un entorno artificial configurable y se observa su iteración con el

⁸⁴ En [11] se encuentran ejemplos de las simulaciones realizadas en este programa.

entorno.

En [11] se encuentran los vídeos con la simulación del comportamiento de los robots desarrollados en esta aplicación, se puede observar la mejora en el desplazamiento de los genotipos debido a los distintos procesos evolutivos a los que han sido sometidos los prototipos.

Marco Socio-Económico

6.1 Planificación

Este trabajo se inició el día 20 de Febrero del 2014, desde el principio se fijaron las etapas de desarrollo del trabajo, en la Tabla 6.1-1 se indican las distintas tareas planificadas.

La primera tarea corresponde al estudio de los Algoritmos Genéticos, gracias a diversos trabajos y ensayos sobre el tema se pudo iniciar en el estudio de este campo así como el análisis de su alcance e impacto en el trabajo.

En la segunda tarea, la más importante, se realizó un estudio detallado de la herramienta y sus distintos módulos. Si bien el manual de Framsticks ha servido de apoyo durante esta fase, la investigación en profundidad ha sido posible gracias a las numerosas pruebas realizadas y a otros trabajos sobre la plataforma encontrados en Internet. En esta etapa se adquirieron todos los conocimientos necesarios para el desarrollo de las siguientes tareas.

La tarea más compleja, ha sido la tercera, el diseño del modelo, en este punto se han realizado numerosos ensayos hasta conseguir el modelo ideal; que se utiliza en la tarea cuatro. Ha sido el cuello de botella del trabajo debido a los diversos problemas que se han encontrado para diseñar un modelo que cumpliera los objetivos del trabajo. Se evaluaron dos posibles opciones y se eligió la solución que más satisfacía los criterios establecidos.

Todas las pruebas realizadas y su análisis se han llevado a cabo en la tarea cuatro. Se considera la más importante, al ser la encargada de verificar que los objetivos se han cumplido mediante el estudio de los resultados obtenidos en la ejecución de esta tarea.

Por último se incluye una tarea adicional encargada de la elaboración de la memoria del trabajo.

Tabla 6.1-1 Planificación del trabajo

Id	Tareas	Nombre de la tarea
1	Tarea 1	Estudio de AG
2	Tarea 2	Estudio de la Herramienta
3	Tarea 3	Diseño del modelo
4	Tarea 4	Pruebas y análisis de resultados
5	Tarea 5	Elaboración memoria de trabajo

En la siguiente Figura 6.1-1, se muestra el diagrama del proyecto con el tiempo estimado que se emplea en la ejecución de cada tarea.

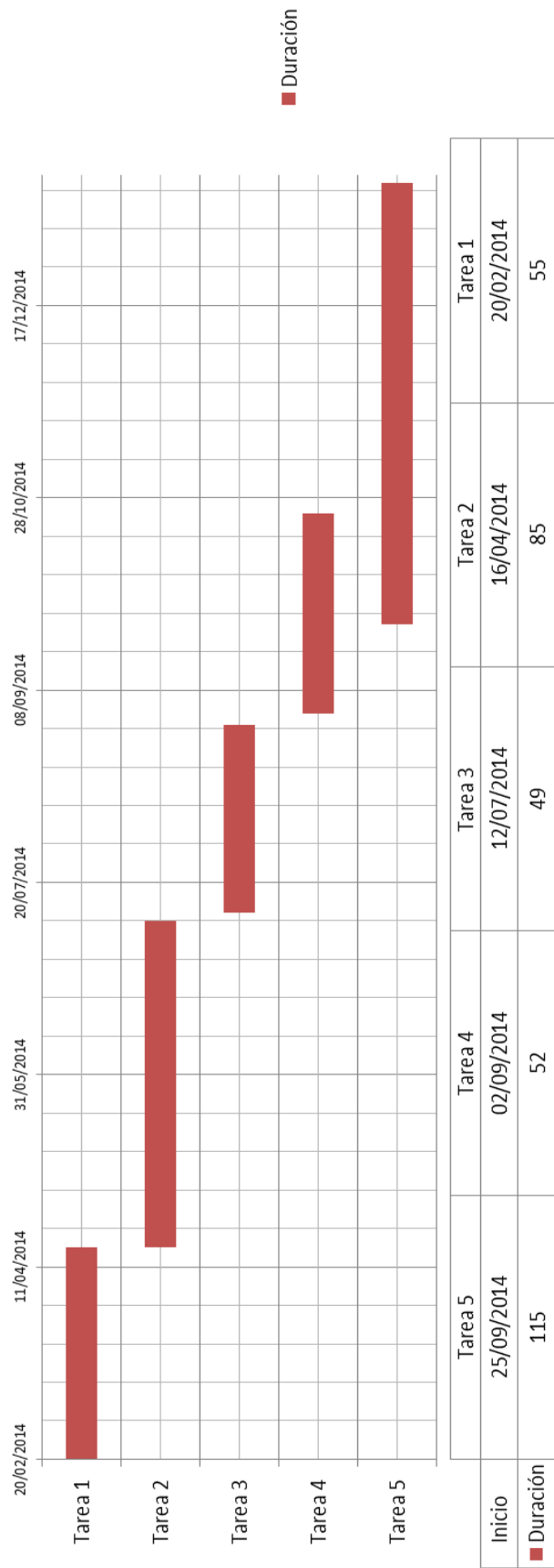


Figura 6.1-1 Diagrama de Gantt

6.2 Presupuesto

A continuación se hace un presupuesto de los costes que tiene la elaboración de este TFG, tanto a nivel personal como de costes de software y hardware, se detalla en la Tabla 6.2-1, Tabla 6.2-2 y Tabla 6.2-3 [12] y [13].

Tabla 6.2-1 Coste Personal

Coste Personal			
Categoría	Nº Horas	Coste hora	Coste Total
Jefe de proyecto	48	35 €	1680 €
Ingeniero Junior	400	10 €	4000 €
TOTAL			5680 €

Para calcular los costes tanto de software como de hardware se hace uso de la siguiente fórmula (Ecuación 6.2-1):

$$\text{Costes} = \left(\frac{A}{B}\right) * C * D$$

Ecuación 6.2-1 Fórmula de Costes

A= Número de meses desde la fecha de facturación en que el equipo es utilizado.

B= Periodo de depreciación.

C= Coste del equipo (sin IVA)

D= Porcentaje de uso dedicado al proyecto. Se fija al 100%.

Tabla 6.2 -2 Costes de Software y Hardware

Descripción	Coste	Dedicación(meses)	Periodo de depreciación (meses)	Coste Imputable
Ubuntu 14.0	0 €	6	60	0 €
Open Office	0 €	6	60	0 €
Framsticks	0€	12	60	0 €
FREDitor	0€	12	60	0 €
Artificial Life	0€	12	60	0 €
Adobe Reader	0 €	10	60	0 €
Costes Hardware				
Portátil Sony Vaio	700 €	12	60	140 €
TOTAL				140 €

Tabla 6.2-3 Costes totales

Costes totales	
CONCEPTO	Coste Imputable
Coste Personal	5680,0 €
Costes Software	0 €
Costes Hardware	140 €
Gastos Generales (20% del Coste personal)	1136,0€
Subtotal 1	6956,0€
Beneficios Empresariales (20% del Subtotal 1)	1391,2 €
Subtotal 2	8347,2€
Coste del riesgo (10% del Subtotal 2)	834,72€
Base Imponible	9181,92€
IVA(21% Base Imponible)	1928,20€
TOTAL	2847,12 €

Capítulo 7

7.1 Conclusiones

Llegados a este punto, es necesario recordar los objetivos planteados y el análisis de los resultados obtenidos. El cumplimiento del objetivo principal se basaba en la documentación de la plataforma Framsticks y el desarrollo de un conjunto de experimentos relacionados con la robótica evolutiva en individuos que imiten a los modelos naturales. Una vez realizado el trabajo, se puede decir que se ha cumplido el objetivo aunque con algunas dificultades y modificaciones.

El primer problema fue la falta de documentación de la herramienta, lo que ocasionó numerosas pruebas de causa-error para averiguar el funcionamiento de todos los elementos de la herramienta y sus limitaciones.

Superada esta fase, el segundo problema, y gran dificultad del proyecto, la creación de un modelo robótico que imitara al comportamiento animal. Hubo bastantes problemas en el diseño de un modelo que se cumpliera este requerimiento; se descartaron varias propuestas pero finalmente se consiguió el modelo definitivo que ha sido el objeto de análisis de este trabajo. Este ha sido el principal cuello de botella, junto con la falta de documentación, del trabajo.

Diseñado el modelo, se ejecutaron las simulaciones que lo evolucionarían hacia el genotipo más óptimo. En este punto se encontró otro inconveniente, la limitación de la evolución, los nuevos genotipos no evolucionaban las variables establecidas en los objetivos. Se realizó una investigación al respecto y se descubrió el concepto de la genética, encarga de fijar los parámetros evolutivos.

Acotado el problema, se propusieron dos modelos evolutivos, evolución parcial y evolución completa. Analizados los resultados de ambos experimentos se puede afirmar que se han cumplido los objetivos fijados.

Se ha realizado un estudio detallado y minucioso de la herramienta, permitiendo la realización de los experimentos requeridos y cuyos resultados se ha comprobado que cumplen los criterios establecidos.

Además del análisis crítico del trabajo experimental, considero importante hacer un análisis a nivel personal. La realización de este trabajo ha supuesto un gran reto, hasta ahora desconocido, la capacidad de afrontar problemas por sí sólo de un tema nuevo y sin conocimiento previo, sin el respaldo de compañeros y profesores.

Es cierto que los tutores de los proyectos juegan un papel importante al guiar al alumno por el camino correcto, pero es él el encargado de resolver las dificultades que se encuentran en ese camino y de buscar las soluciones necesarias para continuarlo.

En ocasiones este camino no es fácil y supone momentos de frustración y desesperación pero es en esos momentos, donde más se aprende y se pone a prueba uno mismo.

7.2 Trabajos Futuros

La herramienta Framsticks, es un programa con un enorme potencial; con la correcta documentación y algunas modificaciones relacionadas con la creación de los modelos se pueden realizar estudios muy interesantes y detallados.

En relación al trabajo con la plataforma, se podría desarrollar una aplicación “puente” para poder utilizar modelos diseñados en otros programas. Con esta aplicación se reduciría los problemas surgidos en la creación de modelos robóticos así como la globalización y alineación con el resto de plataformas relacionadas con la robótica.

De cara a Framsticks sería interesante la creación de nuevos sensores y neuronas para que el usuario pueda reproducir con la mayor exactitud posible, su modelo robótico real, abaratando costes de hardware y tiempo empleado en el proceso evolutivo.

La implementación de nuevos universos para incrementar las funcionalidades de la plataforma es otra de las ideas que se podrían desarrollar. Los nuevos universos supondrían nuevos retos para los robots que tendrían que aprender a desenvolverse en distintos entornos; esto aumentaría sus posibilidades de adaptación al mundo real.

Para el diseño del modelo robótico considero necesario la creación de un módulo dedicado exclusivamente a la simulación del comportamiento del sistema de control global del genotipo, en el que se pueda observar y cuantificar las variaciones en el comportamiento de los genotipos detalladamente; aunque actualmente se encuentra implementado en la GUI una opción similar pero está falta de información y funcionalidades. Este nuevo módulo supondría la reducción del cuello de botella que ha surgido durante la realización de este trabajo, el diseño del modelo robótico.

En cuanto al experimento desarrollado, como trabajo futuro se podrían realizar pruebas utilizando otros tipos de simulaciones; que realicen análisis de aspectos del proceso evolutivo que no han sido abarcados en este trabajo. Por ejemplo experimentos que seleccionen a los modelos que mejor se adapten al entorno, ensayos que comparen varias simulaciones simultáneas.

Glosario

TFG	<i>Trabajo de Fin de Grado</i>
FRED	<i>Framsticks Editor</i>
AL	<i>Artificial Life</i>
AG	<i>Algoritmo Genético</i>
OG	<i>Operador Genético</i>
GUI	<i>Graphical User Interface</i>
SO	<i>Sistema Operativo</i>
ADN	<i>Ácido Desoxirribonucleico</i>
CLI	<i>Command Line Interface</i>

Referencias

- [1] Sims, Karl. *Envolving Virtual Creatures*. Computer Graphics, Annual Conferences Series, Julio 1994; pp 15-22. Disponible [Internet]:
<<http://www.karlsims.com/papers/siggraph94.pdf>> [4 de Abril del 2014]
- [2] *Cyberbotics*. Disponible [Internet]: <http://www.cyberbotics.com/overview> [9 de Septiembre de 2014]
- [3] Lewin, Benjamin. *Genes*. Edición 2ª. Editorial Reverte. 1996. ISBN 84-291-1844-6; pp: 70-72.
- [4] Komosinski, Maciej. Ulatowski, Szymon. *Framstick Manual* Disponible [Internet]: www.Framsticks.com [9 de Septiembre de 2014]

[5] Basogain, Olabe, Xabier. *Redes Neuronales Artificiales y sus aplicaciones*. Escuela Politécnica Superior de Bilbao.2008. Disponible [Internet]: http://cvb.ehu.es/open_course_ware/castellano/tecnicas/redes_neuro/Course_listing.html [8 de Noviembre del 2014]

[6] Koza, R, John. *Genetic Programming On the Programming of Computers by Means of Natural Selection*. Edición 6ª. MIT Press.1998. ISBN 0-262-11170-5.pp: 62-73. Disponible[Internet]: <http://www.ru.lv/~peter/zinatne/ebooks/MIT%20-%20Genetic%20Programming.pdf> [30 de Abril del 2014]

[7] Shettleworth J, Sara. *Cognition Evolution and Behavior*. Oxford University Press. 1998. pp: 176-180.

[8] Bluck, John. NASA 'Evolutionary' software automatically designs antenna. NASA NEWS.2014.Disponible [Internet]: http://www.nasa.gov/mission_pages/st-5/main/04-55AR.html [7 de Diciembre del 2014]

[9] Gestal, Pose, Marcos. *Introducción a los Algoritmos Genéticos*. 2000. Disponible: [Internet] <<http://sabia.tic.udc.es/mgestal/cv/aaggtutorial/aagg.html>> [8 de Junio de 2014]

[10] B. Melián Batista, J.A. Moreno Pérez, J.M. Moreno Vega. *Algoritmos Genéticos. Una visión práctica*. 2009 .Disponible: [Internet] <http://www.sinewton.org/numeros/numeros/71/Darwin_03.pdf> [10 de julio de 2014]

[11] Cabañas, Marina. Canal de Youtube <<https://www.youtube.com/channel/UC-j3dk8ogxLOOsw2X-Ild0g/feed>> [10 de Enero del 2015]

[12] Serrano Simón, Tamara “IDENTIFICACIÓN DE SISTEMAS DINÁMICOS MEDIANTE TEST DE RELÉ MODIFICADO”.Universidad Carlos II de Madrid.2014.

[13] Leal Díaz, Oscar. “OBTENCIÓN, TRATAMIENTO Y VISUALIZACIÓN DE ESTADÍSTICAS MEDIANTE HERRAMIENTA WEB”. Universidad Carlos II de Madrid.2012

[14] Růžicka, Vít. “VIRTUAL CRIATURES SIMULATED IN FRAMSTICKS”. CzechTechnical University de Praga. Primavera 2014. Disponible [Internet]: <https://cyber.felk.cvut.cz/research/theses/papers/493.pdf> [30 de Junio del 2014]

[15] Ruiz Reina, Jose L. *Inteligencia Artificial*, 2014. Disponible [Internet]: <<http://www.cs.us.es/cursos/iati/temas/tema-05.pdf>> [16 de Julio de 2014]

[16] Komosinski, Maciej. *The Framsticks system: versatile simulator of 3D agents and their evolution*. Kybernetes: The International Journal of Systems & Cybernetics, pp: 156–173, 2003. Disponible [Internet]: <http://www.framsticks.com/files/common/Komosinski_FramsticksSystem_Kybernetes2003.pdf>. [15 de Septiembre del 2014]

[17] Komosinski, Maciej. Theory in Biosciences. Diciembre 2001, pp: 271-286. Disponible [Internet]:

<http://www.framsticks.com/files/common/Komosinski_Similarity_TheoryInBiosc2001.pdf>

[10 de Octubre del 2014]

[18] Rechenberg, Ingo. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Fromman-Holzboog Verlag, 1973

Anexos

Modelos

A. FLINK

A.1. Código

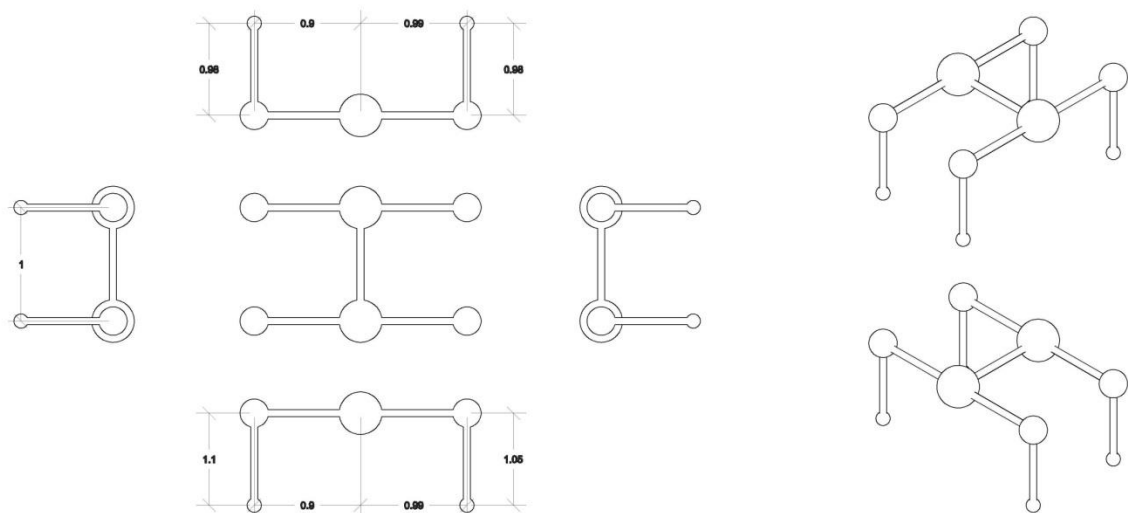
```
org:
name:FLINK
genotype:~
//0
p: 1.0, m=3.0, ing=0.15625, as=0.15625
p: 1.0002964, -0.99999994, m=2.0, ing=0.175, as=0.175
p: 0.3880819, -1.5003524, -0.6122425, fr=1.12, ing=0.19, as=0.19
p: 2.0, m=3.0, ing=0.123239, as=0.123239
p: 2.0002964, -0.99999994, m=2.0, ing=0.148592, as=0.148592
p: 2.6128073, -1.4999895, -0.6122425, ing=0.168873, as=0.168873
p: 2.0002964, 0.99999994, m=2.0, ing=0.148592, as=0.148592
p: 2.6128073, 1.4999895, -0.6122425, ing=0.168873, as=0.168873
p: 1.0002964, 0.99999994, m=2.0, ing=0.175, as=0.175
p: 0.3880819, 1.5003524, -0.6122425, fr=1.12, ing=0.19, as=0.19
j: 0, 1, rx=0.7853, rz=-1.5705, dx=1.0, stam=0.175
j: 1, 2, rz=-1.047, dx=1.0, stam=0.17
j: 0, 3, dx=1.0, stam=0.123239
j: 3, 4, rx=-0.7853, rz=-1.5705, dx=1.0, stam=0.148592
j: 4, 5, rz=1.047, dx=1.0, stam=0.168873
j: 3, 6, rx=0.7853, rz=1.5705, dx=1.0, stam=0.148592
j: 6, 7, rz=-1.047, dx=1.0, stam=0.168873
j: 0, 8, rx=-0.7853, rz=1.5705, dx=1.0, stam=0.175
j: 8, 9, rz=1.047, dx=1.0, stam=0.19
n: p=0, d="N"
n: d="*"
n: p=3, d="N"
n: j=2, d="@: 0.6"
n: p=3, d="N"
n: j=2, d="|: 0.630282, 0.333333"
n: j=4, d="@: 0.6"
n: j=8, d="@: 0.6"
n: p=4, d="N: 0.8, 0.04, 2.0, 0.0"
n: p=8, d="N: 0.8, 0.04, 2.0, 0.0"
n: p=3, d="*: "
n: j=6, d="@: 0.25"
n: p=6, d="N: 0.8, 0.04, 2.0, 0.0"
```

```

n: j=1, d="@: 0.25"
n: p=1, d="N: "
c: 0, 1, -0.2
c: 0, 4, -2.89
c: 0, 2, -1.808
c: 2, 0, 1.849
c: 2, 2, 3.094
c: 2, 2, -1.387
c: 3, 2
c: 4, 2, 1.287
c: 4, 4, 0.5
c: 5, 4
c: 6, 8
c: 7, 9, 0.136
c: 8, 1
c: 8, 10
c: 9, 1
c: 9, 10
c: 11, 12
c: 12, 10
c: 13, 14
c: 14, 1
~

```

A.2. Plano esquemático



A.3. Longitud de los nexos

Tabla Anexo-1 Medidas de FLINK

Articulaciones	Nexo	Distancia
1,2	1	1
1,0	0	1
0,8	7	1
8,9	8	1
0,3	2	1
4,3	3	1
3,6	5	1
6,7	6	1
4,5	4	1

B. IKKI

B.1. Código

```
org:
name:IKKI
genotype:~
//0
m:
p:1.0, m=3.0, ing=0.15625, as=0.15625
p:1.00029632679056, -0.999999956095216, m=2.0, ing=0.175, as=0.175
p:0.388081886162751, -1.50035246820048, -0.612242470200107, fr=1.12,
ing=0.19, as=0.19
p:2.0, m=3.0, ing=0.123239, as=0.123239
p:2.00029632679056, -0.999999956095216, m=2.0, ing=0.148592, as=0.148592
p:2.6128071955969, -1.49998954926428, -0.612242470200107, ing=0.168873,
as=0.168873
p:2.00029632679056, 0.999999956095216, m=2.0, ing=0.148592, as=0.148592
p:2.6128071955969, 1.49998954926428, -0.612242470200107, ing=0.168873,
as=0.168873
p:1.00029632679056, 0.999999956095216, m=2.0, ing=0.175, as=0.175
p:0.388081886162751, 1.50035246820048, -0.612242470200107, fr=1.12,
ing=0.19, as=0.19
j:0, 1, rx=0.7853, rz=-1.5705, dx=1.0, 0.0, 0.0, stam=0.175
j:1, 2, rz=-1.047, dx=1.0, 0.0, 0.0, stam=0.17
j:0, 3, dx=1.0, 0.0, 0.0, stam=0.123239
j:3, 4, rx=-0.7853, rz=-1.5705, dx=1.0, 0.0, 0.0, stam=0.148592
j:4, 5, rz=1.047, dx=1.0, 0.0, 0.0, stam=0.168873
j:3, 6, rx=0.7853, rz=1.5705, dx=1.0, 0.0, 0.0, stam=0.148592
j:6, 7, rz=-1.047, dx=1.0, 0.0, 0.0, stam=0.168873
j:0, 8, rx=-0.7853, rz=1.5705, dx=1.0, 0.0, 0.0, stam=0.175
j:8, 9, rz=1.047, dx=1.0, 0.0, 0.0, stam=0.19
n:p=0
n:d=*
n:p=3
n:j=2, d="@: 0.6"
n:p=3
n:j=2, d="|: 0.630282, 0.333333"
```



```

n:j=4, d="@: 0.6"
n:j=8, d="@: 0.6"
n:p=4, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=8, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=3, d="*: "
n:j=6, d="@: 0.25"
n:p=6, d="N: 0.8, 0.04, 2.0, 0.0"
n:j=1, d="@: 0.25"
n:p=1, d="N: "
c:0, 1, -0.2
c:0, 4, -4.525
c:0, 2, -1.808
c:2, 0, 1.849
c:2, 2, 3.094
c:2, 2, -1.387
c:3, 2
c:4, 2, 1.287
c:4, 4, 0.5
c:5, 4
c:6, 8
c:7, 9, 0.136
c:8, 1
c:8, 10
c:9, 1
c:9, 10
c:11, 12, 1.599
c:12, 10
c:13, 14
~

```

C. IKKI'15

C.1. Código

```

org:
name:IKKI'15
genotype::~
//0
m:
p:0.57, -0.158, -0.117, m=3.0, ing=0.15625, as=0.15625
p:0.982, -0.98, 0.709, m=2.0, ing=0.175, as=0.175
p:0.5990000000000003, -1.544, -0.5469999999999997, fr=1.12, ing=0.19, as=0.19
p:2.174, 0.014, 0.292, m=3.0, ing=0.123239, as=0.123239
p:2.00029632679056, -0.99999995609521, -1.11022302462516e-16, m=2.0,
ing=0.148592, as=0.148592
p:2.26, -1.375999999999999, -0.533, ing=0.168873, as=0.168873
p:2.018, 0.9150000000000001, 0.385, m=2.0, ing=0.148592, as=0.148592
p:2.427, 1.432, -0.6439999999999997, ing=0.168873, as=0.168873
p:0.638, 1.333, -0.119, m=2.0, ing=0.175, as=0.175
p:0.546, 1.714, -0.776, fr=1.12, ing=0.19, as=0.19
j:0, 1, rx=0.7853, rz=-1.5705, dx=0.822122050547978, 0.875197187113638,
0.293000371452262, stam=0.175
j:1, 2, rx=-4.05707551305453e-18, -1.56125112837913e-17, -1.047,
dx=1.28565349015915, -0.091416760524, -0.61729982920529, stam=0.17
j:0, 3, dx=1.604, 0.172, 0.409, stam=0.123239
j:3, 4, rx=-0.7853, rz=-1.5705, dx=1.01394843852377, 0.0834033254131066, -
0.329522876858017, stam=0.148592

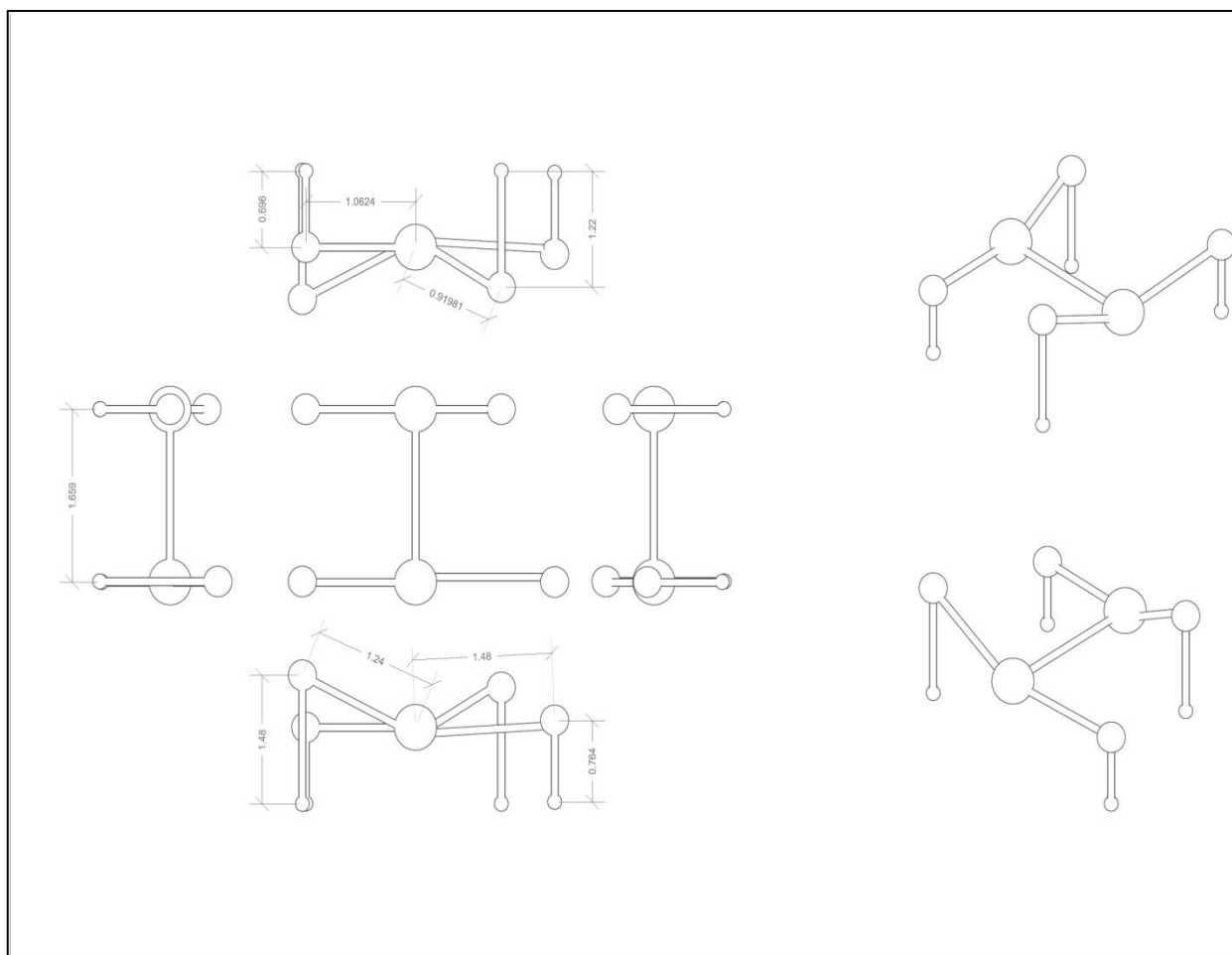
```

```

j:4, 5, rx=4.07722297233376e-18, -1.63985578588433e-17, 1.047,
dx=0.673392668109875, -0.0453450091571405, -0.19338349354399, stam=0.168873
j:3, 6, rx=0.7853, rz=1.5705, dx=0.900953733462463, 0.176262765060696, -
0.0447192107847288, stam=0.148592
j:6, 7, rx=-4.07722297233376e-18, -1.63985578588433e-17, -1.047,
dx=1.13900906684611, -0.0607190185591633, -0.43861434817794, stam=0.168873
j:0, 8, rx=-0.7853, rz=1.5705, dx=1.49102008475972, -0.046361457183677, -
0.0491805055983549, stam=0.175
j:8, 9, rx=4.05707551305453e-18, -1.56125112837913e-17, 1.047,
dx=0.649201338571337, -0.0649720072475839, -0.399487497015122, stam=0.19
n:p=0
n:d=*
n:p=3
n:j=2, d="@: 0.6"
n:p=3
n:j=2, d="|: 0.630282, 0.333333"
n:j=4, d="@: 0.6"
n:j=8, d="@: 0.6"
n:p=4, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=8, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=3, d="*: "
n:j=6, d="@: 0.25"
n:p=6, d="N: 0.8, 0.04, 2.0, 0.0"
n:j=1, d="@: 0.25"
n:p=1, d="N: "
c:0, 1, -0.2
c:0, 4, -4.525
c:0, 2, -1.808
c:2, 0, 1.849
c:2, 2, 3.094
c:2, 2, -1.387
c:3, 2
c:4, 2, 1.287
c:4, 4, 0.5
c:5, 4
c:6, 8
c:7, 9, 0.136
c:8, 1
c:8, 10
c:9, 1
c:9, 10
c:11, 12, 1.599
c:12, 10
c:13, 14
~

```

C.2. Plano esquemático



C.3. Longitud de los nexos

Tabla Anexo-1 Medidas IKKI'15

Articulaciones	Nexo	Distancia
1,2	1	1.48
1, 0	0	1.24
0,8	7	1.48
8,9	8	0.76
0,3	2	1.66
4,3	3	1.06
3,6	5	0.92
6,7	6	1.22
4,5	4	0.69

D. IGOR''5

D.1. Código

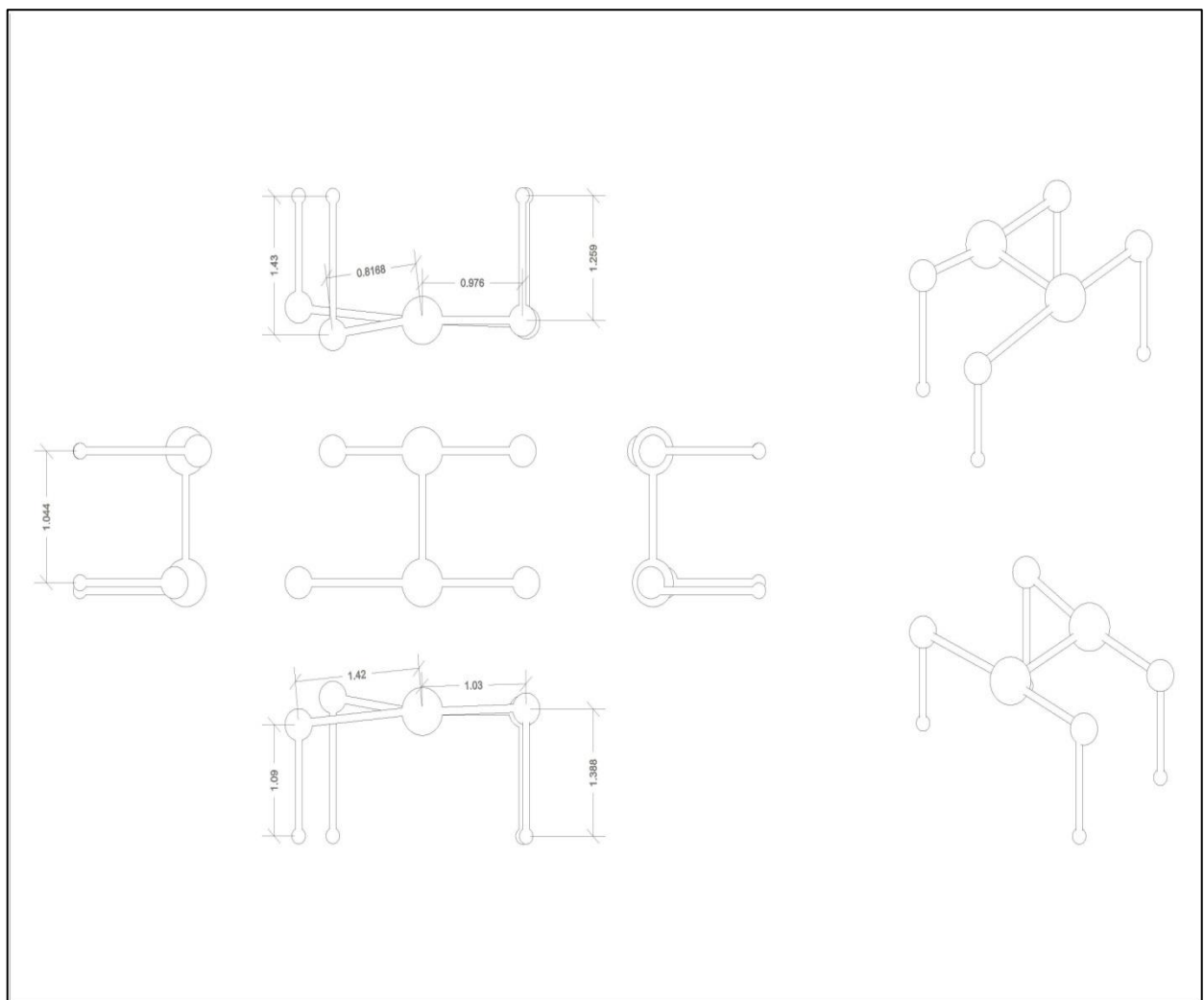
```
org:
name:IGOR''5
genotype::~~
//0
m:
p:1.0, m=3.0, ing=0.15625, as=0.15625
p:1.00029632679056, -0.999999956095216, -1.16621169397021e-19, m=2.0,
ing=0.175, as=0.175
p:0.1740000000000002, -1.498, -0.6179999999999998, fr=1.12, ing=0.19, as=0.19
p:2.02, -0.132, -0.265, m=3.0, ing=0.123239, as=0.123239
p:2.00029632679056, -0.999999956095215, -3.88578058618805e-16, m=2.0,
ing=0.148592, as=0.148592
p:2.699, -1.723, -1.047, ing=0.168873, as=0.168873
p:1.78, 0.81, 0.0629999999999999, m=2.0, ing=0.148592, as=0.148592
p:2.6128071955969, 1.49998954926429, -0.612242470200108, ing=0.168873,
as=0.168873
p:1.514, 0.874, 0.218, m=2.0, ing=0.175, as=0.175
p:0.5239999999999999, 1.414, -0.6570000000000001, fr=1.12, ing=0.19, as=0.19
j:0, 1, rx=0.7853, rz=-1.5705, dx=1.0, -6.59042109889479e-17,
6.57263624443944e-17, stam=0.175
j:1, 2, rx=-4.05707551305453e-18, -1.56125112837913e-17, -1.047,
dx=1.13341196307885, -0.0798505442971212, 0.147291805345484, stam=0.17
j:0, 3, dx=1.02, -0.132, -0.265, stam=0.123239
j:3, 4, rx=-0.7853, rz=-1.5705, dx=0.86799407925962, -0.2014807638122,
0.173288599460948, stam=0.148592
j:4, 5, rx=4.07722297233376e-18, -1.63985578588433e-17, 1.047,
dx=1.43047395802838, -0.00892149508719837, -0.246555366542104, stam=0.168873
j:3, 6, rx=0.7853, rz=1.5705, dx=0.941928840211959, 0.40182793511704,
0.0620674676181501, stam=0.148592
j:6, 7, rx=-4.07722297233376e-18, -1.63985578588433e-17, -1.047,
dx=1.26850317089013, 0.0644023491286724, 0.111165820116859, stam=0.168873
j:0, 8, rx=-0.7853, rz=1.5705, dx=0.874152273597566, -0.517439540039903, -
0.20906966534507, stam=0.175
j:8, 9, rx=4.05707551305453e-18, -1.56125112837913e-17, 1.047,
dx=1.41199481418347, 0.192316933786344, 0.0813009329467513, stam=0.19
n:p=0
n:d=*
n:p=3
n:j=2, d="@: 0.6"
n:p=3
n:j=2, d="|: 0.630282, 0.333333"
n:j=4, d="@: 0.6"
n:j=8, d="@: 0.6"
n:p=4, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=8, d="N: 0.8, 0.04, 2.0, 0.0"
n:p=3, d="*: "
n:j=6, d="@: 0.25"
n:p=6, d="N: 0.8, 0.04, 2.0, 0.0"
n:j=1, d="@: 0.25"
n:p=1, d="N: "
c:0, 1, -0.2
c:0, 4, -2.89
c:0, 2, -1.808
c:0, 0
c:2, 0, 1.849
c:2, 2, 3.094
```

```

c:2, 2, -1.387
c:3, 2
c:4, 2, 5.179
c:4, 4, 0.5
c:5, 4
c:6, 8
c:7, 9, 0.136
c:8, 1
c:8, 10, -0.476
c:8, 8
c:9, 1, -0.207
c:9, 10
c:9, 2
c:11, 12, 0.627
c:12, 10
c:12, 12, -2.642
c:13, 14
c:14, 1, 1.395
~

```

D.2. Plano esquemático



D.3. Longitud de los nexos

Tabla Anexo-3 Medidas de IGOR''5

Articulaciones	Nexo	Distancia
1,2	1	1.09
1, 0	0	1.42
0,8	7	1.03
8,9	8	1.39
0,3	2	1.04
4,3	3	0.82
3,6	5	0.98
6,7	6	1.26
4,5	4	1.43